Syntactic Annotation of the Hebrew CHILDES Corpora

Shai Gretz

Syntactic Annotation of the Hebrew CHILDES Corpora

Research Thesis

Submitted in Partial Fulfillment of The
Requirements for the Degree of Master of Science in
Computer Science

Shai Gretz

Submitted to the Senate of the Technion - Israel
Institute of Technology

Sivan, 5773 Haifa May 2013

The research thesis was done under the supervision of Prof. Alon Itai (Department of Computer Science, Technion) and Prof. Shuly Wintner (Department of Computer Science, University of Haifa) in the Department of Computer Science.

The general financial help of the Technion is gratefully acknowledged.

Contents

List of Figures

List of Tables

Abstract

The CHILDES database is a large collection of child—adult spoken interactions in over 25 languages. Automatic annotation of these data facilitates research on child language development and acquisition by providing researchers with a large amount of accurate data. Recently, the English section of the CHILDES database was automatically annotated with labeled dependency relations in a state-of-the-art approach. We describe a similar endeavor, focusing on the Hebrew section of CHILDES. This is done by the following process: First, we design a novel annotation scheme of dependency relations reflecting constructions of child and child-directed utterances, as well as the special phenomena of the Hebrew language. We then annotate a corpus with these dependency relations, and use the manually-annotated data to train a parser with which the rest of the corpora can be annotated. We then evaluate the parsing accuracy. We show the adaptability of our annotation scheme to the CHILDES corpora in numerous evaluation scenarios. We also examine different annotation approaches of linguistic issues relevant to several languages or unique to Hebrew, as well as the contribution of morphological features to the accuracy of dependency parsing of the Hebrew section of CHILDES. This is the first syntactic parser of Hebrew spoken language.

Chapter 1

Introduction

1.1 The CHILDES Database

Child language development and acquisition has long been a topic of interest amongst researchers from different areas of study, from linguistics to psychology. The CHILDES database (MacWhinney, 2000), consisting of child—adult spoken transcripts from various languages, has been an enormous help for this line of research. The uniqueness of this database is that it provides its users not only with raw data from monologic, dyadic and multi-party interactions (all following a unified and established transcription scheme) but also with tools for the application of theoretically-motivated and well-tested analyses at the various levels of linguistic structure and use occurring in these transcripts.

Until recently, researchers who wanted to explore a certain hypothesis regarding acquisition and distribution of a given grammatical pattern, resorted to manually annotating parts of the CHILDES corpora according to their needs. This is a laborious method since the manual annotation takes time and effort and furthermore, collaborative work is harder to maintain as each researcher makes his own manual annotation for his specific needs. However, Sagae et al. (2010) developed a parser for automatically annotating the entire English section of the CHILDES database. First, 18,863 utterances (roughly 65,000 words) were manually and automatically annotated. The corpus was used for training a data-driven parser which was then tested on an independent corpus and used to annotate the entire English section of CHILDES. Sagae et al. (2010) report about 94 percent accuracy when evaluating the parser on one of the corpora in the English section of CHILDES. Following the syntactic annotation of the English section of

CHILDES, other works were commenced with the goal of annotating more sections in different languages in CHILDES.

Our work concentrates on automatic syntactic parsing of the Hebrew section of the CHILDES database. The Hebrew section consists of five corpora, and our work focuses on two of these that had their transliteration unified and were also morphologically annotated (Albert et al., forthcoming) — the Berman longitudinal corpus (Berman and Weissenborn, 1991) and the Ravid longitudinal corpus. These two corpora range in speaker age and in size, with approximately 110,000 utterances in total (Nir et al., 2010).

Hebrew has a problematic orthography, and many different words can appear orthographically identical, since vowels are represented partially (if at all). However, due to the fact that the transcription of these corpora is vocalized, vowels and stress are expressed explicitly allowing the rich morpho-phonological structure of Hebrew to differentiate between these possible interpretations and thus to be sensitive to the correct pronunciation and meaning of the words in question in context. Wherever ambiguity still remains, a manual disambiguation process was applied in order to yield a fully disambiguated morphological analysis. Thus, the parsing of the Hebrew section of CHILDES is based on corpora with full, reliable, and disambiguated Part-of-Speech morphological annotation.

1.2 Dependency Grammar

Dependency grammars are a theoretical framework for syntactic representation of a sentence that puts the emphasis on functional relations (such as *subject* and *object*) between pairs of words. This is in contrast to the representation of constituent grammars, where the emphasis is on creating structural units (such as noun phrase or verb phrase) using derivation rules. The first works to present dependency grammars as a linguistic theory in the modern era date back to early-mid 1900's (see Percival (1976)). Since then, many theoretical frameworks for dependency grammars were established (Kübler et al., 2009).

In dependency grammars, a sentence is represented by a graph. The nodes of the graph are the tokens of the sentence — most commonly words, but sometimes morphemes (e.g., Eryiğit et al. (2008)). The directed arcs of the graph connect pairs of tokens. These connections are known as *dependency relations* (also known as *functional* or *grammatical relations*, for example `Subj` for a subject relation). In each aforementioned pair one token is denoted as the *head* of the relation and the other is denoted as the *dependent* (the modifier or the complement of the head). Each token in the sentence, except the root, has one head and may serve as the head of multiple dependents, thus forming a tree. The relations between words may be *labeled*, where for each edge the exact syntactic type of the relation is stated, or *unlabeled*, where no type is stated.

Dependency grammars seem adequate for our line of work for a number of reasons. Typologically, Hebrew is a language with relatively complex morphology and flexible constituent structure, annotation schemes based on dependency grammars proved to be successful with data-driven parsing algorithms and for languages with relatively free word order and high morphological ambiguity. However, it should be noted that this is not true for every morphologically rich language (see sections 1.3 and 1.4). More importantly, dependency grammars seem more suited than constituent grammars when describing early stages of language acquisition. Whereas constituent grammars based analyses may be suitable for structural linguistic complexities, dependency grammar frameworks can successfully account for some of the characteristics of child language development in Hebrew, namely attachments of immediate neighboring words, and head-first attachments (Ninio, 1996, 1998). Furthermore, dependency structures directly specify the name of the functional relation, which is important for querying a certain type of grammatical function — as opposed to constituent trees where the type of relation can be derived from a more complex set of constituents such as NP or VP. But despite the fact that a dependency grammar allows specifying a direct dependency relation, it still maintains a tree hierarchy of the sentence.

Figure 1.1 depicts an example of a dependency structure of an English sentence taken from Sagae et al. (2010). The direction of the edges is from the head to its dependents. For example, 'eat' is the head of 'We', with a relation labeled `Subj` (for subject), and of 'sandwich', with a relation labeled `Obj` (for object).

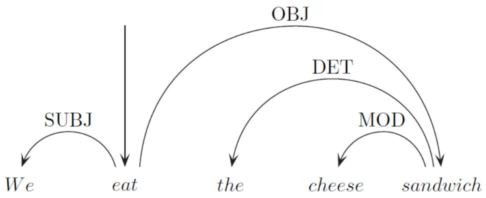

Figure 1.1: An example of a dependency structure.

As stated above, the sentence is represented as a tree and there is one token which does not have a head — the *root*. For the purpose of ensuring that all the tokens are dependent in some relation, a special `Root` marker is added upon which the main predicate of the sentence (hence, the root of the tree) is dependent. In the example above, the vertical arrow pointing to 'eat' represents the relation of 'eat' with the `Root`.

In the CHILDES database and throughout this work, dependency structures are represented as follows: each token is marked with a triplet $i|j|REL$, where i is the index of this token in the utterance (starting from 1), j is the index of the head of the current token (the special `Root` marker is given the index 0), and REL is the label of the relation between them.

Following is the example of Figure 1.1 in our representation:

(1.1)	*We*	*eat*	*the*	*cheese*	*sandwich*
	`1|2|Subj`	`2|0|Root`	`3|5|Det`	`4|5|Mod`	`5|2|Obj`

For example, the analysis 1|2|Subj under the token 'We' means that the token 'eat' is the head of 'We' in a `Subj` relation, and 3|5|Det under 'the' means that the token 'sandwich' is the head of 'the' in a `Det` relation.

1.3 Dependency Parsing

Dependency parsing algorithms assign a dependency structure to their input sentences. Dependency parsing as a syntactic parsing method has been used in several works over the past decade. Those works have shown that dependency parsing yields good results in a variety of languages, which are typologically very distinct (Kübler et al., 2009).

One of the main developments in dependency parsing is the use of *data-driven models*: algorithms that use ideas and methods derived from the field of supervised machine learning. These algorithms use classifiers trained on sentences annotated with correct (manually tagged) dependency relations. Thus there are two stages when using these algorithms — *learning* from trained data to create a classifier and *parsing* a given sentence using the trained classifier.

One type of data-driven parsing is *transition-based* parsing: the parsing process is modeled as a transition system between parser states. Transition-based algorithms are generally inspired by the deterministic shift-reduce algorithm (Aho and Ullman, 1972). This algorithm uses two data structures: a *queue* holding tokens yet to be processed (initialized to hold the entire sentence in order) and a *stack* holding partially processed tokens. The current state of the parser is usually determined by these data structures and by the set of relations created until this point. At each step the token in front of the queue is either pushed on top of the stack (an action referred to as *shift*), or is attached to the token on top of the stack in some direction and with some label (*reduce*). In the context of transition-based parsers, the action to be performed at each step is predicted by the trained classifier, given the current state of the parser. The algorithm terminates when the queue is empty (Kübler et al., 2009).

Various dependency parsers are available for use. `The Stanford Parser` (de Marneffe et al., 2006) initially had generated phrase structures that later were also converted to allow output of dependency relations. The conversion process is based on identifying the (semantic) heads of the constituents of the phrase structure tree, and then identifying their respective dependencies. At the second phase the type of relations are identified where the most specific relation type possible is selected.

`EasyFirst` (Goldberg, 2011) is a parser recently developed in which the sentence is not traversed from left to right when being parsed but rather from the easiest connections to the more complex ones in a bottom-up manner. When evaluating this parser on English it outperformed MaltParser.

`MaltParser` (Nivre et al., 2006) is an architecture of transition-based parsers that can support different learning and parsing algorithms, each accompanied with its feature set and parameters. The majority of algorithms that MaltParser supports are inspired by the shift-reduce algorithm described above. The feature set upon which the SVM classifier is trained is derived from the surface forms, the base forms and the morphological information of a subset of the tokens in the data structures (i.e., the queue and the stack) that comprise the state of the parser. MaltParser allows costumizing the feature set used for training the classifier, as well as other parameters.

`MEGRASP` (Sagae and Tsujii, 2007) is another transition-based parser whose parsing process is based on the shift-reduce algorithm. The algorithm is a dependency version of the data-driven constituent parsing algorithm for probabilistic GLR-like parsing described by Sagae and Lavie (2006). The action to be performed at each parsing step is predicted by a maximum entorpy classifier trained on correctly annotated data sets. The maximum entropy classifier sets for each parsing action a probability based on the features that represent the state of the parser. The states that can be derived from the current state are all kept in a heap sorted by their probabilities. The probability of a state is the product of the probabilities of the parsing actions that lead to this state. The next state to be explored is chosen in a best-first manner — the state with the highest probability. MEGRASP was used for dependency parsing of the English section of CHILDES (Sagae et al., 2010).

For evaluation (see chapters 4 and 5) we use MaltParser and MEGRASP. MEGRASP is used because it was used for annotating the English corpora of CHILDES and because it is compatible with the format of the CHILDES database. MaltParser is used as a state-of-the-art transition-based parser that also allows configuring different parameters (such as the learning algorithm, the parsing algorithm and the feature set) relatively easily.

1.4 Related Work

Dependency parsing is most commonly preferred over constituent grammar-based parsing in languages that have rich morphology and relatively free word order (Kübler et al., 2009), and one such language is Czech. The Prague Dependency Treebank (Hajič et al., 2001) is a treebank of Czech corpora annotated with three layers of annotation — morphological, analytical (dependency relations) and tectogrammatical (a higher level of annotation of underlying relations which do not appear in the analytical annotation). For our purposes the analytical layer is of interest when discussing possible annotation decisions (see section 6.1). The analytical annotation specifies a tree representation of the sentence where each token is given an attribute called *analytical function*. The analytical function is the relation of the token with its head token in the tree — e.g., `Sb` for subject, `Obj` for object (Hajičová et al., 1999).

Another work worth noting concerns the Arabic language, in particular the Prague Arabic Dependency Treebank (Hajič and Zemánek, 2004). Arabic was one of toughest languages to parse in the 2007 CoNLL (Conference on Computational Natural Language Learning) shared task, where accuracy scores of dependency parsing in Arabic were among the lowest of the ten languages tested in the task (Nivre et al., 2007). Dependency parsing of Arabic is interesting in our context given that Arabic is a Semitic language that has many common features with Hebrew, mainly rich morphology, deficient orthography and relatively free word order. We refer to the work on the Prague Arabic Treebank when we introduce our representation of tokens in section 2.2.2. The Prague Arabic Treebank maintains the same levels of annotation that are maintained in the Czech Dependency Treebank described above. It is worth noting, that despite the general advantage of dependency parsing over constituency parsing in morphologically rich languages, constituency parsers for Arabic that proved to be more competitive have been developed (Green and Manning, 2010).

Marton et al. (2013) present an elaborate analysis of the effect that inflectional and lexical features have on parsing Arabic. They systematically evaluated different feature set configurations derived from gold and predicted (i.e., non-gold) morphology when parsing parts of the Columbia Arabic Treebank (Habash and Roth, 2009). They used MaltParser for their evaluation. Their work is relevant to section 5.4 where we examine the contribution of morphological features to the accuracy of parsing of the Hebrew section of CHILDES.

Another interesting work has been done on Turkish (Eryiğit et al., 2008). The work on Turkish work is interesting due to the level of representation of words these researchers chose to implement. Turkish is distinctively agglutinative and rich in head-final constructions (where the dependent precedes the head in a pair of related tokens) (Eryiğit et al., 2008). Due to the agglutinative nature of the Turkish language, the dependency scheme is based on relations between different morphemes (referred to as *inflectional groups* or IGs) rather than between words. The work on Turkish showed that creating a dependency tree based on the representation of the sentence using IGs instead of whole words yields better accuracy when parsing Turkish sentences. This representation is probably more appropriate to the unique phenomena of Turkish described above. Hebrew does not exhibit this kind of agglutination so we feel that the representation chosen for Turkish is not adequate for Hebrew. However, Hebrew does show fused forms (e.g., inflected prepositions and suffixed nouns for possession) for which we make some use of separate morpheme representation, rendering the representation of tokens partially morpheme-based (see section 2.2.2).

To the best of our knowledge, there are no significant annotated resources of spoken Hebrew. However, there are annotated resources of *written* Hebrew. The main resource is the Hebrew constituent structure treebank (Sima'an et al., 2001), a relatively small corpus which contains around 6200 sentence taken from an Israeli daily newspaper.

A more recent work dealing with *written* Hebrew was done by Goldberg (2011). His work produces two types of structures — constituent structures and dependency relations — generated from the Hebrew constituent structure treebank. Besides the EasyFirst parser described above, his work also presents a novel annotation scheme for the syntactic annotation of written Hebrew. Note that the his current scheme leaves some relations unlabeled and others — for example `SUBJ`, `OBJ` and `COMP` — labeled. We design a fully labeled annotation scheme. His annotation is different from ours due to a number of reasons:

1) His scheme is based on written, journalistic Hebrew. CHILDES is comprised of spoken, colloquial Hebrew.

2) His scheme is based on written adult Hebrew. The CHILDES database is comprised of child—adult spoken interactions, specifically interactions between adult non-expert speakers and children who are still acquiring their grammar.

In our analysis of linguistic issues (see section 6) we discuss some of the ideas presented in Goldberg (2011). In section 2.2.1 we further describe the characteristics of the CHILDES corpus which distinguishes it from other corpora.

1.5 Contributions

This work makes several contributions:

1) We define the first annotation scheme for spoken Hebrew based on dependency grammar.

2) We develop a parser for the Hebrew section of the CHILDES database, annotating utterances with syntactic dependency relations.

3) We evaluate our parser in different domains and compare different annotation schemes that are linguistically plausible.

Thus, our main research focus is building a parsing model for spoken Hebrew using a novel annotation scheme, and parsing the entire Hebrew section of CHILDES. A secondary research goal is to evaluate variations to our original scheme empirically, as well evaluation of Hebrew morphological input to the learning process.

Chapter 2

Hebrew CHILDES Characteristics

In this section we present the Hebrew corpus available from CHILDES. We explain the way the data is organized and the challenges that were faced as a result of Hebrew-specific phenomena and spoken language characteristics.

2.1 Levels of Data

The corpora in the CHILDES database are comprised of three levels of data, each refered to as a *tier* (MacWhinney, 2000):

1. The *main* tier, which contains actual utterances.
2. The *mor* tier, which contains disambiguated lexical and morphological analyses of the utterances (for languages that have a morphological analysis).
3. The *gra* tier, where the syntactic analyses of the utterances is to be inserted.

The data are organized in a one-to-one format, where every token in the main tier has exactly one counterpart in the mor tier and the gra tier.[1] The examples that appear throughout this work present these three tiers. The majority of the examples given in the following sections are taken directly from the Hebrew CHILDES corpus. For ease of reading, the examples are presented in the following format:

[1] This is true except in special cases where a token was marked to prevent it from receiving a mor analysis (see section 2.2.1).

<Actual utterance (main tier)> <English gloss>
<Morphological analysis (mor tier)>
<Syntactic analysis (gra tier)>
<Full utterance translation>

Example 2.1 shows an utterance with all three tiers. The elements in each of the three tiers are vertically aligned.[2]

(2.1) *ʔat “you”* *mešaqēret “lie”*
`pro:pers|gen:fm&num:sg` `part|gen:fm&num:sg`
`1|2|Aagr` `2|0|Root`
“You are lying!”

We now move to elaborate on the content of each respective tier. The actual utterance in the main tier is presented in the same transliteration as in the CHILDES corpora (see Nir et al. (2010)). The utterances in the corpora include an utterance-final punctuation mark. For brevity we omit this punctuation from the examples presented in this work.

The morphological analysis in the mor tier (Albert et al., forthcoming) presents the part of speech (POS) and other morphological attributes taken from the mor tier of the CHILDES corpus.[3] Each token that is represented in the mor tier also recieves an analysis in the gra tier. The analysis in the mor tier begins with the part of speech (POS) of the token. The rest of the morphological data is separated from the POS tag by the | sign. Each morphological attribute is of the form <feature:value>, the ‘feature’ being the morphological feature (e.g., ‘gen’ for gender, ‘num’ for number, ‘pers’ for person) and the ‘value’ being its value (e.g., ‘fm’ for feminine gender, ‘sg’ for singular number, ‘1’ for first person). Pairs of <feature:value> are separated from one another with an ‘&’ sign. For example, the word *mešaqēret* “lie” in Example 2.1 is analyzed as ‘part’ (participle), followed by ‘gen:fm’ (gender: feminine) and ‘num:sg’ (number: single). The mor tier is fully disambiguated automatically using a module presented in Albert et al. (forthcoming). The corpora used for our evaluation (see section 4) was also disambiguated manually.

The syntactic analysis in the gra tier presents the dependency structure of the utterance in the format explained in section 1.2 (see Example 1.1).

[2] Throughout this work, in case the actual utterance is split into more than one line due to space constraints, the other tiers (except the full sentence translation) are also split and each line of analysis follows its corresponding part of utterance (e.g., Example 3.2).

[3] Note that the actual mor tier contains additional data which are not relevant here.

For example, the word ʔat "you" in Example 2.1 is attached to the word *m*ešaqēret "lie" with the label `Aagr` (Agreeing argument, see section 3.2.1).

2.2 Unique Constraints of the CHILDES Corpus

2.2.1 Spoken Language

One relevant feature of the corpus is the fact that we are dealing with spoken language — and in particular with language spoken by children who are still acquiring their grammar. The study of spoken language entails dealing with situations where utterances are partial as part of what we could see as processing constraints. Spoken language is also characterized with repetitions, repairs, interruptions and utterances that may contain more than one clause. Some utterances may be rendered as ungrammatical, especially those uttered by children.

The transcriptions attempt to reflect the flow of the conversation as accurately as possible, and so the format of the corpora allows insertion of symbols with tokens that are to be ignored for the purpose of morphological and syntactic analyses, for example in the case of false starts or repetitions where the beginning of the utterance should be ignored (MacWhinney, 2000). [4]

For example, consider the following utterance:

<table>
<tr><td>(2.2)</td><td><i>ʔem "em"</i></td><td><i>ma "what"</i></td><td>, ","</td><td><i>[//]</i></td><td><i>mi "who"</i></td><td><i>ʔomēr "say"</i></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td><code>que</code></td><td><code>part</code></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td><code>1|2|Aagr</code></td><td><code>2|0|Root</code></td></tr>
</table>

<table>
<tr><td><i>le- "to"</i></td><td><i>ʔat "you"</i></td><td><i>bōʔi "come"</i></td><td><i>? "?"</i></td></tr>
<tr><td><code>prep</code></td><td><code>pro:person</code></td><td><code>v</code></td><td><code>?</code></td></tr>
<tr><td><code>3|2|Anonagr</code></td><td><code>4|3|Aprep</code></td><td><code>5|2|Anonagr</code></td><td></td></tr>
</table>

"Who says to you 'come'?"

The first two tokens and the comma are considered a false start using the [//] symbol and thus do not get morphological and syntactic analyses.

Other cases may include words which are not recognized by the morphological analyzer, due to an irregular form or inconsistent transcriptions. Albert et al. (forthcoming) report that when the grammar built for this corpora is applied to a new corpus, 1.75 percent of the tokens do not receive an

[4] However, the transcriptions are not consistent with respect to these markings. Corrections have been made on these corpora throughout the work where possible but still problematic instances may remain.

analysis. All these properties of the corpora, resulting from the effect that spoken language has on the transcription, might have a negative impact on the quality of parsing.

On the other hand, there are some properties of the transcription which derive from the fact that we are dealing with spoken language and may improve the quality of parsing. The fact that we are dealing with spoken language allows the transcriptions to be vocalized, with vowels and stress expressed explicitly as discussed in section 1.1. The vocalization leaves very little morphological ambiguity. For the purposes of the corpora used in this thesis, this ambiguity is subsequently resolved in a manual process, thus it is entirely morphologically disambiguated. For processing new data, an automatic disambiguation process takes place. Albert et al. (forthcoming) show that the automatic morphological disambiguation module achieves an accuracy of 96.6 percent. This is in contrast to written Hebrew which suffers from high morphological ambiguity. Furthermore, spoken language implies shorter utterances (as witnessed by Sagae et al. (2010)) that are thus generally simpler and easier to annotate.

2.2.2 Token Representation

Representation of tokens could be crucial for successful syntactic analysis. As discussed in section 1.4, the work of Eryiğit et al. (2008) showed how the representation of bound morphemes improved syntactic accuracy of Turkish. Hebrew is rich in bound morphology, and tokens that consist of more than a single morpheme are quite common. Given the original transcription and token representation of the Hebrew CHILDES corpora (see Albert et al. (forthcoming)), the main issue that needs to be addressed regarding the representation of the data is whether bound morphemes are represented in the main tier or not. While Hebrew is not agglutinative as Turkish, it does present some phenomena that may require morpheme-bound representation. Thus, we decided to split certain word types to allow partial morpheme-bound representation of the data. This is done using a pre-processing script designed to modify the representation of certain word types, as described below, and split them to represent the relevant morphemes.

Our motivation for these splits is mostly computational — to reduce data sparseness and improve parsing accuracy. Certain fused forms may rarely appear, possibly making it more difficult for the parser to identify them and analyze them correctly. The split of fused forms to separate tokens, isomorphic to those that appear in non-fused forms, may reduce the likelihood that the parser did not see this form before and is not able to recognize it and its part in the construction. It is important to note that these changes are only for the syntactic analysis, and following its completion we merge the split tokens back together, omitting any inner-construction relations. We thus avoid sparseness by splitting these fused tokens before parsing, and we also maintain the one-to-one representation of the corpora by fusing these tokens back together after parsing without losing important syntactic information.

Before handling fused morphemes, we explain how the transcription handles a specific group of tokens that are orthographically adjacent to the subsequent word in standard Hebrew. Hebrew represents a specific case where there are orthographically adjacent tokens that are considered separate morphemes in other languages, and in spoken Hebrew seem to have word-like status (Nir and Berman, 2010) — these include four prepositions (e.g., *be-* "in"), the conjunction *we-* "and" as well as the definite article (*ha-* "the"). These appear orthographically adjacent to the subsequent word in written Hebrew, whereas in the representation of CHILDES they are transcribed as separate tokens (e.g., *be- māyim* ('in water')). On the other hand, Multi-lexemic expressions that are typically written as separate tokens are treated as a single token in our transcription and should be analyzed as such by the parser.

We now move to the types of fused forms we chose to split. One type of fused morphemes that were treated as separate tokens are prepositions fused with a definite article. In prepositional phrases, some simplex prepositions are phonologically (and, subsequently, orthographically) fused with the definite article (*ba-* "in the" for *be-* "in" *ha-* "the", *la-* "to the" for *le-* "to" *ha-* "the", *ka-* "like the" for *ke-* "like" *ha-* "the"). In our corpora these forms are considered as separate tokens (e.g., *ba-* "in the" is split to two tokens, one for *be-* "in" and one for *ha-* "the").

Fused morphemes occur in Hebrew not only with the definite article but also with inflected prepositions (e.g., *ʕim* "with" and its inflected form *ʔitāḳ* "with you", *bišvīl* "for" and its inflected form *bišvīlēḳ* "for you", *šel* "of" and its inflected form *šelāh* "hers") and the accusative marker (e.g., *ʔet* "ACC" and its inflected form *ʔotō* "him"). These forms occur only when the pronoun is not a grammatical subject, and exhibit irregularities depending on the case and the type of the preposition (e.g., compare the locative *ʕlav* "on him" and the accusative ʔoto "to him"). In the original transcription, these fused morphemes are considered as one token in all tiers. Similarly to what was done for fused, definite prepositions, we treat these fused morphemes as multiple tokens and split them accordingly into two tokens, one representing the base preposition and the other representing the pronoun it is inflected for (e.g., *itāḳ* "with you" is split into *ʕim* "with" *ʔat* "you", *bišvīlēḳ* "for you" is split into *bišvīl* "for" *ʔat* "you"). As stated above, the non-fused version of prepositions (whether orthographically adjacent in standard Hebrew to the subsequent word or not) are treated as a separate token, and thus the split construction is isomorphic to it.

The same decision was taken with respect to possessive suffixes of nouns. We split the fused morphemes into three tokens — one representing the base noun, one representing the possessive marker *šel* "of" and one represeting the suffixed pronoun it is inflected for. For example, the possessive inflection of *ʔaxot* "sister", *ʔaxotī* "my sister", is split into *ʔaxot* "sister" *šel* "of" *ʔani* "I". Note that we do not claim these cases contain 3 morphemes but rather 3 *sememes* (semantic units).

Another construction in Hebrew that requires pre-processing is the double genitive, marked both with a suffix and with a subsequent possessive marker (Netzer and Elhadad, 1998). These constructions are relatively rare in our corpora. They are split according the same guidelines as for suffixed nouns. For example, consider the sentence *ze* "it" *sofō* "his-end" *šel* "of" *kol* "all" *balōn* "baloon" — "This is the end of every balloon". *sofō* "his-end" is inflected and the split sentence is *ze* "it" *sof* "end" *šel* "of" *huʔ* "he" *šel* "of" *kol* "all" *balōn* "balloon".

It is relevant to compare these representations to the ones in Arabic, as expressed in the Prague Arabic Dependency Treebank (Hajič and Zemánek, 2004). We present the decisions that were made for Arabic for illustration purposes as it was not part of our decision making process. Like Hebrew, Arabic also exhibits a definite article and some prepositions and conjunctions that attach to the subsequent noun, as well as inflected nouns with possessive suffixes and verbs marked with the accusative case. The morphological analysis of the Prague Arabic Treebank is based on the idea of splitting words into *morphs* or *segments*, each receiving its own morphological analysis (Smrž and Pajas, 2004). Prepositions (e.g., *bi-* "in"), conjunctions (e.g., *wa-* "and") and suffixed pronouns (e.g., *-ha* "her" in *zawju-ha* "her husband") are all considered separate tokens (i.e., morphs) in the representation of the data (e.g., *biduni* "without me" is split into *bi- duni*), even if they are orthographically attached in Modern Standard Arabic script. Note that nouns with possessive suffixes are split into two tokens — one representing the noun and the other representing the pronominal suffix — as opposed to our representation of three (e.g., *ʔaxotī* "my sister" is split into *ʔaxotī* "sister" *šel* "of" *ʔani* "I"). However, the definite article *al* "the" remains attached to the noun it modifies and does not appear separately (e.g., *al-muhandisu* "the engineer" remains the same). A definite tag is marked in the morphological analysis of such words thus differentiating it from the indefinite form.

After understanding the characteristics of spoken Hebrew and their effect on the transcription, and after dealing with the representation of the tokens, we can now move to present the annotation scheme — the group of relations used to syntactically annotate the corpora.

Chapter 3

Annotation Scheme for Hebrew CHILDES

In this section we present a dependency annotation scheme for Hebrew CHILDES utterances.

3.1 General Framework

The scheme is defined in terms of independent utterances (in other words, no inter-utterances relations are accounted for).

Our scheme is inspired by the scheme of the English section of CHILDES (Sagae et al., 2010) as some relations are defined similarly. The English scheme was comprised of 37 relations. This adaptation is done mostly for issues that are general for spoken language and not unique to Hebrew spoken language. For example, coordination constructions are a challenge for syntactic annotation in general and for annotating these constructions we decided to use the definition of the English scheme, where the head of a coordination construction is the coordinating conjunction and the dependents are the coordinated elements, a relation labeled `Coord`. Also, we took into consideration the work done by Goldberg (2011) on dependency parsing of written Hebrew, specifically in Chapter 6 where we evaluate alternative approaches for specific relations.

Recall that dependency grammar is based on the distinction between heads and dependents. We distinguish between three types of dependents: arguments [A], modifiers [M] and others. *Arguments* are subcategorized dependents of the heads that they modify: typically, they are semantically required by the head, their properties are determined by the head, and they can occur at most once (often, exactly once).[1] *Modifiers*, on the other hand, are non-subcategorized dependents: typically, they select the head that they depend on (in the sense that they specify the properties of the head they depend on rather than the other way around), and, consequently, they may occur zero or more times. The *Others* group contains functional relations in which the dependents do not necessarily complement or modify their heads, or relations in which the dependents do not relate specifically to any other token in the utterance. For example, the `Com` label marks the relation in which a communicator is the dependent. A communicator is generally related to the entire utterance, but to maintain the dependency structure we mark the main predicate of the utterance as the head of the communicator.

Typically in dependency grammar, the root of an utterance is an inflected verb or a copula in verbless copula utterances, carrying the tense marking in the clause. In utterances where the copula is elided and there is no element carrying a tense, the head is the predicating element. Note that analyses of copula sentences may prefer to mark the predicating element as the head even when the copula is not elided (see sections 3.2.2 and 6.1). When an utterance is lacking any of the above, the root is the element on which other elements depend (such as the noun with respect to its modifiers). In single word utterances, the token is by default the root.

An important issue that needs to be addressed is the fact that we are dealing with transcriptions combined of child speech and child directed speech. One could suggest an alternative approach where a separate scheme is developed for each type of speaker. However, we did not feel the need to devise such a separation, considering the transcriptions we have at hand. Child utterances did pose a challenge to manual annotation, particularly with neologisms and repetitions. Comparing the child speech and the child-directed speech of the corpora used for evaluation revealed that child-speech contains about 1.5 times more repetitions and multiple times more uses of the Unk (unknown) label, used to mark detached unknown relations (see section 3.2.6). This definitely had an effect on the difficulty of annotating and parsing child speech, but did not seem to necessitate a separate relation scheme.

[1] We use the terms ‘argument’ and ‘complement’ interchangeably throughout this document.

3.2 Dependency Relations

The annotation scheme is comprised of 24 basic dependency relations and a few more complex dependency relations (a concatenation of two basic dependency relations; see section 3.2.5). We discuss below some of the main constructions covered by our scheme; the full taxonomy of the annotation scheme is listed in Table 3.1 and in more detail in Appendix A.

Arguments	**Modifiers**	**Others**
Aagr	Mdet	Voc
Anonagr	Madj	Com
Aprep	Mpre	Coord
Ainf	Mposs	Srl
Acop	Mnoun	Enum
Aexs	Madv	RelCl
	Mneg	SubCl
	Mquant	Unk
	Msub	Punct

Table 3.1: Taxonomy of labels.

3.2.1 Agreeing and Non-agreeing Arguments

In Hebrew, arguments of verbs can either agree with the verb or not. The type of argument is defined by the verb and the semantic function of the argument. Typically, a verb has at most one agreeing argument whereas the number of non-agreeing arguments a verb takes can be zero or more. The agreement features are number, gender and person, morphological information that appears in the mor tier (see section 2.1). The relation between a verb and an agreeing argument is labeled `Aagr` and the relation between a verb and a non-agreeing argument is labeled `Anonagr`. The standard terminology for the former is *subject* and for the latter is *object*. The reason for choosing formal labels as opposed to functional labels is to remain consistent and to avoid theory-specific controversies.

In example 3.1, the verb **dag** is the head of its agreeing argument **hu?** with the `Aagr` relation and of its non-agreeing argument **dagi̅m** with the `Anonagr` relation. Notice the agreement in number between **hu?** and **dag** (both are single)[2] and the non-agreement between **dag** and **dagi̅m** (the latter is plural).

[2] Note that the participle in Hebrew is not inflected for person, so the agreement here

(3.1)	**huʔ** *“he”*	**dag** *“fish”*	**dagīm** *“fish”*
	`pro:person|num:sg&gen:ms`	`part|num:sg&gen:ms`	`n|num:pl`
	`1|2|Aagr`	`2|0|Root`	`3|2|Anonagr`
	“I don’t want drops.”		

3.2.2 Copular elements and Existential Markers

Copular elements and existential markers, as well as other forms of the verb *hayā* “be”, are discussed elaborately in section 6.1[3]. Copular elements and existential markers are most commonly inflections of the verb *hayā* “be” in the past and future tenses and the imperative mood. In the present tense, copular elements take the form of pronouns and can sometimes be omitted, whereas the existential marker in the present tense is most commonly *yeš* “there_is” (or its negative counterpart *ʔeyn* “there_is_not”) and is also sometimes optional. There are two linguistically plausible approaches when annotating a construction containing a copula — the first is when the copula is the head and the second is when its nominal or adjectival argument is the head (these two approaches apply to other forms of *hayā* as well). In section 6.1 we evaluate both of these approaches empirically. Since copular elements and existential markers are syntactically different, we mark them with different labels. To mark the relation between a copula and its argument we use the relation `Acop` for the first linguistic approach (the copula is the head) and `Xcop` for the second linguistic approach. To mark the relation between an existential marker and its argument we use the relation `Aexs` for the first linguistic approach (the existential marker is the head) and `Xexs` for the second linguistic approach.

3.2.3 Clauses

Clauses can be either arguments or modifiers of a verb or a noun. As an argument, a clause can be headed by either a non-finite verb or a complementizer. If it is headed by a non-finite verb, the relation between the verb in the main clause and the non-finite verb is labeled `Ainf`. If it is headed by a complementizer, the relation between the verb in the main clause and the complementizer is labeled `Anonagr` and the relation between the complementizer and the main predicate in the subordinate clause is labeled `SubCl`.

holds for number and gender.

[3] The inflected form of *hayā* “be” may also fill the role of an auxiliary, but those were extremely rare in our corpora and thus are not treated separately.

In Example 3.2, **laʕavōr** is the non-finite argument of the main verb **titēn**.

(3.2)

ʔaz “so”	**titēn** *“you-let”*	*le- “to”*	*ʔanī “I”*	**laʕavōr** *“pass”*
`adv`	`v`	`prep`	`pro:person`	`v`
`1|2|Com`	`2|0|Root`	`3|2|Anonagr`	`4|3|Aprep`	`5|2|Ainf`

“So let me pass.”

If there is no comlpementizer, the predicate of the subordinate clause is dependent on the predicate of the main clause directly with the `Anonagr` relation, such an in Example 3.3, where the verb **qarā** — the predicate of the subordinate clause — is directly dependent on the verb **tirʔī**.

(3.3)

ʔoy “oh_no”	**tirʔī** *“see”*	*ma “what”*	**qarā** *“happen”*	*le- “to”*	*hiʔ “she”*
`co`	`v`	`que`	`v`	`prep`	`pro:person`
`1|2|Com`	`2|0|Root`	`3|4|Aagr`	`4|2|Anonagr`	`5|4|Averb`	`6|5|Aprep`

“Oh no, look what happened to her.”

When the clause is a modifier of a verb or a noun, the relation between the verb or noun and the complementizer is labeled `Msub` (Modifying subordinated clause). If the clause is a relative clause, the relation between the relativizer and the main predicate of the relative clause is labeled `RelCl`.

3.2.4 Compund Nouns and Construct Phrases

A common phenomenon in Hebrew is that of construct phrases (referred to in Hebrew as ‘smiḳut’) which often indicate possession amongst other meanings (Glinert, 1989). A construct phrase is comprised of two nouns where the first is in the construct state and the second noun modifies it. The relation between the two nouns is labeled `Mnoun`. Note that when the two nouns comprise a single idiomatic unit they are marked as one token in the transcription.

In Example 3.4 **xelqēy matēḳet** is a construct phrase and thus the relation between them is labeled `Mnoun`.

(3.4) *we-* *“and”* *bifnīm* *“inside”* *yeš* *“there_is”* **xelqēy** *“part”*
`conj` `adv` `exs` `n`
`1|0|Root` `2|3|Madv` `3|1|Coord` `4|3|Aexs`
matēḳet *“metal”*
`n`
`5|4|Mnoun`
“And inside there are parts of metal.”

3.2.5 Elision Relations

Processing spoken language includes dealing with missing elements, whether as a result of true ellipsis or interruptions and incomplete utterances. In the English section of CHILDES, Sagae et al. (2010) decided to mark the missing elements as elided and to relate to them in the analysis using *elision relations*. These relations are a combination of two basic relations: one marking the relation with which an existing element would have been dependent on the elided element; and the other marking the relation with which the elided element would have been dependent on another element. Following the scheme for English, we also mark missing elements with elision relations. Elision relations are unambiguous for human annotators, but inflate the annotation scheme to some extent and may be difficult for the parser to identify because of their rarity. In the corpora used for evaluation (see Chapter 4.1) 22 elision relations were introduced.

In Example 3.5 **ha-** is marked with the `Mdet-Aprep` relation. `Mdet` stands for the relation between **ha-** and an elided element and `Aprep` stands for the relation that would have been marked between the elided element and the preposition *leyād*.

(3.5) *leyād* *“near”* **ha-** *“the”*
`prep` `det`
`1|0|Root` `2|1|Mdet-Aprep`
“Near the -”

3.2.6 Child Language

As the CHILDES corpus is comprised of child and adult interactions, child-specific forms and constructions are rather frequent. These include neologisms, babbling, and incoherent speech. Such forms may be detached from the utterance. They can either be labeled with the `Unk` relation which marks unknown relations, such as in Example 3.6; or, when the syntactic function of such forms is known to the annotator, they can take the place of a known relation (e.g., the neologism *bdibiyabi* in Example 3.7.

(3.6)	**curu** *“curu”*	*gam “also”*	*lecayēr “paint”*
	chi	adv	v
	1|3|Unk	2|3|Madv	3|0|Root

“??? also to paint.”

(3.7)	*ʕḳšā̆yw “now”*	*ʔanī “I”*	*holēḳet “walk”*	*le- “to”*
	adv	n|num:sg	part|num:sg	prep
	1|3|Madv	2|3|Aagr	3|0|Root	4|3|Anonagr
	bdibiyabi *“bdibiyabi”*			
	chi			
	5|4|Aprep			

“Now I am going to bdibiyabi.”

Chapter 4

Parsing

In this section we present the data on which we evaluate the adaptation of the annotation scheme to the CHILDES corpora. We introduce the different evaluation scenarios we run and the parsers we use for the evaluation process.

4.1 Data

The data we use for evaluation consist of files taken from the Hebrew CHILDES section. The data were annotated by two lexicographers; all disagreements were resolved by a third annotator, a linguist who specializes in syntactic analyses.

We manually annotated 12 files — 8 files taken from the Ravid corpus and 4 taken from the Berman corpus — according to the scheme outlined in section 3. The 8 files of the Ravid corpus contain transcriptions of the same child at different ages (ranging from 1;11 to 2;05). The 4 files of the Berman corpus reflect 4 different children (all different from the child in the Ravid corpus) at different ages (2;04, 3;00, 3;03 and 3;06). Statistical data of the corpora are given in Table 4.1. Fused morphemes (such as inflected prepositions and suffixed pronouns) are split (see section 2.2.2). The results and data presented here relate to the corpora after the split and do not include punctuation.

4.2 Parsers

As mentioned in section 1.3 there are several available dependency grammar parsers. We chose to use two of them: MEGRASP (Sagae and Tsujii, 2007) and MaltParser (Nivre et al., 2006). MEGRASP works directly on

		Utterances		Tokens		MLU		MLUw	
Corpus	Files	Total	CS	Total	CS	Total	CS	Total	CS
Ravid	8	4107	1541	13863	3975	3.93	2.93	3.37	2.58
Berman	4	2224	1126	9392	4241	4.9	4.32	4.22	3.77

Table 4.1: Statistics of corpora used for evaluation. MLU is the average number of morphemes per utterance. MLUw is the average number of tokens per utterance. CS is Child Speech.

the CHILDES format in which the corpora are stored, the format that is used to present the examples throughout this work. MaltParser was chosen over the other parsers due to a number of reasons. First, it supports a number of formats, including the CONLL-X shared task format (Nivre et al., 2007). An advantage of using MaltParser is that it also supports costume-made formats, allowing variation in the lexical and morphological information available for the learning algorithm. We used a format similar to CONLL, with the addition of columns representing independent morphological attributes (instead of the concatenated FEATS column). Using MaltParser we examined in section 5.4 the effect of adding morphological features (e.g., number and person) to the default feature set so that the learning algorithm can deduce whether those features are relevant for making syntactic decisions (e.g., enforcing agreement between the main verb and its agreeing argument).

Furthermore, MaltParser supports a variety of configurations for training and parsing. One can select a learning algorithm, parsing algorithm or feature set — amongst others — out of a pre-defined group of options. To achieve the best possible results using MaltParser we used the recently developed *MaltOptimizer* (Ballesteros and Nivre, 2012). MaltOptimizer analyzes the data in a three-phase process and outputs the recommended configuration under which to run MaltParser (e.g., a certain parsing algorithm or a feature set that yield the best results). Throughout the evaluation we used MaltParser with the configuration suggested by MaltOptimizer that was optimized over the training files, unless stated otherwise. We initiated the MaltOptimizer process to select the most appropriate parameters for MaltParser only on the relevant training sets without having access to the test set at each evaluation scenario.

4.3 Evaluation Methodology

Our evaluation consists of two main parts: *In-domain evaluation*, where training is on parts of the Ravid corpus and testing is on other parts of the same corpus (held out during training); and *Out-of-domain evaluation*, where training is done on the files of the Ravid corpus and testing is done on the files of the Berman corpus. We run both MEGRASP and MaltParser on these evaluation scenarios. We also run a 5-fold cross-validation on the Ravid corpora and on both corpora combined. We then examine the effect that an enhanced feature set that includes morphological information can have on the accuracy of parsing. For this purpose we use MaltParser which has a modifiable feature set. In addition, we compare different annotation alternatives that concern a few linguistic issues characteristic to Hebrew. This comparison allows us to see which approach of annotation yields better accuracy.

The evaluation metrics that we use are *unlabeled attachment score* (UAS) and *labeled attachment score* (LAS). In UAS a token is considered correctly annotated if its head is the same head that is marked in the gold-standard — regardless of the grammatical relation. In LAS a token is considered correctly annotated if both the head and the grammatical relation are the same as in the gold-standard. In addition we report *Exact Match* (EXM), the percentage of utterances that are parsed without any errors. These are standard metrics in the evaluation of dependency parsing (Kübler et al., 2009).

To examine the quality of the parsers and the annotation scheme on individual relations, we introduce further metrics which are relation specific — $URecall_r$ (unlabaled recall), $LRecall_r$ (labeled recall), $UPrecision_r$ (unlabeled precision) and $LPrecision_r$ (labeled precision) for some relation r (Kübler et al., 2009). Let us mark $l_g(x)$ as the gold label of token x and $l(x)$ as the label given by the parser. Similarly, let us mark $h_g(x)$ as the head that token x is attached to in the gold file and $h(x)$ as the head that token x is attached to by the parser. The four metrics above are defined as follows:

$$URecall_r = \frac{|\{x \mid l_g(x) = r \wedge h_g(x) = h(x)\}|}{|\{x \mid l_g(x) = r\}|}$$

$$LRecall_r = \frac{|\{x \mid l_g(x) = l(x) = r \wedge h_g(x) = h(x)\}|}{|\{x \mid l_g(x) = r\}|}$$

$$UPrecision_r = \frac{|\{x \mid l(x) = r \wedge h_g(x) = h(x)\}|}{|\{x \mid l(x) = r\}|}$$

$$LPrecision_r = \frac{|\{x \mid l_g(x) = l(x) = r \wedge h_g(x) = h(x)\}|}{|\{x \mid l(x) = r\}|}$$

The first two metrics are a refinement of the recall metric for each relation as the analysis is with respect to the appearances of the relation in the gold standard files. $URecall_r$ is the percent of tokens that are attached with the correct head by the parser out of those that are labeled with the relation r in the gold standard files. $LRecall_r$ is the percent of tokens that are attached with the correct head and labeled with the correct relation by the parser out of those that are labeled with the relation r in the gold standard files. In other words — the percent of tokens of this relation in the gold data that were correctly parsed.

The other two metrics are a refinement of the precision metric for each relation as the analysis is with regards to the appearances of the relation in the parsed model files. $UPrecision_r$ is the percent of tokens that are attached with the correct head by the parser out of those that are labeled with the relation r in the model files. $LPrecision_r$ is the percent of tokens that are attached with the correct head and labeled with the correct relation by the parser out of those that are labeled with the relation r in the model files. In other words — the percent of tokens of this relation in the parsed data that were correctly parsed.

For each of the UAS and LAS precision and recall pairs we calculate the (balanced) f-score, which is the harmonic mean of precision and recall.

In addition to testing the corpus as a whole we show results that relate separately to two types of data: Child Directed Speech (CDS) and Child Speech (CS). On these separate types of data we evaluate the following combinations:

1) Training on both CS and CDS and testing on both CS and CDS (referred to as All-All).

2) Training on CS and testing on CS (CS-CS).

3) Training on CDS and testing on CDS (CDS-CDS).

4) Training on CDS and testing on CS (CDS-CS).

We expect different results when evaluating these four configurations. When training on CS and testing on CS there should be significant changes in the types of utterances and their complexity both in the in-domain evaluation, due to the change of age, and in the out-of-domain evaluation, due to the different children.

When training on CDS and testing on CDS there should be little difference in the same corpus or between corpora since both the training set and the test set contain child directed speech which is pretty much correctly structured in all of the files. However, child-directed speech is known to be tuned to the development of child speech, and the dynamics of this adaptation are an ongoing debate subject. A recently published research based on the English section of CHILDES has shown that the complexity of child-directed speech is strongly correlated to that of child speech and that child-directed speech tends to adapt to within-dialog changes of the child and not only to the child's age and overall development (Kunert et al., 2011). So some changes are expected in this scenario, too.

When training on CDS and testing on CS there may be different types of utterances in the training set and the test set. The utterances in CDS are generally longer and contain relations and structures that do not appear in the CS data. This configuration is interesting because comparing it to the CS-CS configuration can provide insight on the way a child acquires language.

In Table 4.2 we present how the test set in each evaluation scenario was formed. In the in-domain evaluation scenario we build the training set and test set for each of these configurations separately, using 8 files of the Ravid corpus. The files of the Ravid corpus are chronologically ordered by the age of the target child and thus in the in-domain evaluation scenario the held-out set always contains utterances of the same child at an older age. In configurations 1-3, where the data type of the training set and the test set is the same, the training set is comprised of 80 percent of the utterances of the relevant data type in the corpus, holding out 20 percent for the test set. The training set of the All-All configuration contains 3286 utterances (11155 tokens), the CS training set contains 1237 utterances (3246 tokens) and the CDS training set contains 2066 utterances (7946 tokens). The 80 percent of CS (configuration 2) and CDS (configuration 3) are derived from the set of utterances of their respective data types in the corpus, and not from the training set of both data types (configuration 1). This is why the sum of the sizes of the CS and CDS training sets does not necessarily equal the size of the training set of the All-All configuration. In configuration 4 the training set and test set are comprised of utterances of different data types so the entire set of utterances of each data type in the corpus is used, and not just 80 percent of it. This explains why the sizes of the training sets of configurations 3 and 4 are not equal (although both are CDS), as well as the test sets of configurations 2 and 4 (although both are CS).

In the out-of-domain evaluation scenario the training sets and test sets of the different configurations are taken from different sets of files, so the entire set of utterances of the respective data type is used.

	In-domain	Out-of-domain
All-All	20% of the entire Ravid corpus	The entire Berman corpus
CS-CS	20% of the CS of Ravid corpus	The entire CS of Berman corpus
CDS-CDS	20% of the CDS of Ravid corpus	The entire CDS of Berman corpus
CDS-CS	The entire CS of Ravid corpus	The entire CS of Berman corpus

Table 4.2: How the test sets were built.

For all evaluation scenarios, we exclude punctuation and single-token utterances, to avoid artificial inflation of scores.

Chapter 5

Results

In this section we present the results of our evaluation. This section includes an in-domain and out-of-domain evaluation, a cross-validation evaluation (indifferent to domains) and an evaluation of the contribution of morphological features.

5.1 In-domain Evaluation

We first evaluate the parsers on the same domain they were trained on. Table 5.1 shows the accuracy of parsing obtained by MEGRASP and MaltParser. The table shows the number of utterances for the training set and the test set of each of the four configurations, and the UAS, LAS and EXM for both MEGRASP and MaltParser. Note that the results in this table and in the following tables are not directly comparable due to the different number of utterances in each scenario.

				MEGRASP			MaltParser		
Train	Size	Test	Size	UAS	LAS	EXM	UAS	LAS	EXM
All	3286	All	590	87.4	82.3	62.7	91.2	86.6	71.1
CS	1237	CS	183	91.9	87.3	75.4	93.9	89.1	78.1
CDS	2066	CDS	400	85.4	80.8	56.7	89.2	83.9	63.2
CDS	2566	CS	969	84.2	78.5	62.3	88.2	82.5	68.2

Table 5.1: Results: Accuracy of parsing, in-domain.

The results of the in-domain evaluation scenario reveal some interesting details. First, considering the relatively small training set, both parsers achieve reasonable results. Evidently, MaltParser proves to be better than

MEGRASP on this domain. In the All-All, CS-CS, CDS-CDS and CDS-CS configurations MaltParser had better UAS, LAS and EXM. We tested the significance of this difference in the All-All configuration and the advantage of MaltParser over MEGRASP proved to be statistically significant for all three metrics ($p < 0.05$).

To show the contribution of MaltOptimizer, we ran MaltParser with its default parameters. In the All-All configuration the UAS was 84.5 and the LAS was 80.5 — lower both from the results obtained by the optimized MaltParser and from the results obtained by MEGRASP.

Note also the low EXM when testing on CDS as opposed to the high EXM when testing on CS. Recall that the utterances in CDS are longer on average (see Table 4.1) and so there is a higher chance that one of the tokens in an utterance is tagged erroneously.

It is interesting to examine the differences between the results of the CS-CS and CDS-CS configurations. These results may provide insight regarding language acquisition. Higher accuracy in the CS-CS configuration could indicate that the child speech at an older age is more similar to its own speech at a younger age, reflecting self-motivated language development and learning from peers. On the other hand, higher accuracy in the CDS-CS configuration may indicate that the speech of the child at an older age is influenced more by the speech of his caretakers at a younger age.

It appears that for parsing child speech it is better to learn from child speech than from child-directed speech. This is despite the fact that in the CDS-CS configuration the training set is larger. To examine the possiblity that the specific CS test set used in both configurations contributes to this difference, we evaluated the CDS-CS configuration with a training set similar in size to the CS-CS training size (i.e., 1237 utterances) and with an identical test set to the one used in the CS-CS configuration. Table 5.2 shows the results of the modified CDS-CS evaluation (line 2) compared to the CS-CS evaluation (line 1) and the original CDS-CS evaluation (line 3).

				MEGRASP			MaltParser		
Train	Size	Test	Size	UAS	LAS	EXM	UAS	LAS	EXM
CS	1237	CS	183	91.9	87.3	75.4	93.9	89.1	78.1
CDS	1237	CS	183	91.1	85.5	69.9	92.3	88.0	77.6
CDS	2566	CS	969	84.2	78.5	62.3	88.2	82.5	68.2

Table 5.2: Results: Accuracy of parsing, in-domain, CDS vs. CS.

When running the modified CDS-CS configuration, MaltParser yielded an UAS of 92.3, a LAS of 88.0 and an EXM score of 77.6, and MEGRASP yielded an UAS of 91.1, a LAS of 85.5 and an EXM score of 69.9. These are all considerably higher than the original CDS-CS configuration, possibly attributed to this CS test set being easier to parse than the 969 utterances of the test set of the CDS-CS configuration presented in line 3. This could be contributed also to the fact that the test set is taken from the recordings of the child at an older age, thus it is perhaps more similar to CDS data than the CS test set of the original CDS-CS configuration which consists of the entire CS data. The scores of the modified CDS-CS configuration are lower than the CS-CS scores, though the differences are not statistically significant.

The fact that training on CS has some advantage over training on CDS when parsing CS can be partially explained by the fact that the age range of the files of the Ravid corpus is rather small, the difference between the first file and the eighth file being only 7 months. Note that in the CDS-CDS configuration the scores are also relatively low. It is apparent that training on CDS confuses the parser to some degree. This can be explained by the rich structure of CDS compared to CS and by the different constructions and relations uttered by the same adults when the child matures.

We attempted to improve parsing accuracy by constructing an integrated parser. We evaluated MaltParser on the original All-All configuration, but instead of using one parser for the entire data, we used two parsers: one that used a classifier that was trained on the CS data of the training set and one that used a classifier that was trained on the CDS data of the training set. These models were used to parse the CS and CDS data of the test set of the All-All configuration, respectively[1]. Thus the parser that was used integrated two parsers, each built on one data type. However, this method did not provide any improvement. We contribute this possibly to the relatively small size of data from each speech type, or perhaps to the features of this specific corpus.

It is interesting to examine the accuracy of the parsers on specific relations, to see which relations are more difficult for the parsers to predict. Table 5.3 shows the metrics $URecall_r$, $LRecall_r$, $UPrecision_r$ and $LPrecision_r$ for interesting individual relations, as presented in section 4.3, using the transcriptions parsed by MaltParser in the All-All configuration.

Relations that occur with a small group of tokens as dependents (such as

[1] Note that these sub-sets of the test set of the All-All configuration are different from the respective test sets in the CS-CS and CDS-CDS configuration, as explained in section 4.3

	UAS			LAS		
Relation	Recall	Precision	F-score	Recall	Precision	F-score
Root	93.2	92.7	92.9	93.2	92.4	92.8
Aagr	92.1	95.2	93.6	89.5	91.9	90.7
Aprep	99.4	98.2	98.8	99.4	97.6	98.5
Anonagr	94.6	92.6	93.6	89.2	84.3	86.7
Mdet	99.1	98.6	98.8	98.6	98.6	98.6
Madv	87.1	80.4	83.6	78.5	71.6	74.9
Com	74.6	71.2	72.7	65.7	66.7	66.2
Mpre	79.7	75.8	77.7	61.0	58.1	59.5
Aexs	97.6	95.2	96.4	97.6	95.2	96.4
Mquant	78.6	87.5	82.8	50.0	87.5	63.6

Table 5.3: Results: Accuracy of parsing of individual relations, in-domain.

Mdet where the dependent is mainly the token *ha-* "the") or after a specific type of token (such as Aprep that appears after a preposition) achieved a score of 97 percent or above in all the four metrics. The frequent relations Aagr and Root obtained good scores of over 92 percent in URecall and UPrecision and 89 percent in LRecall and LPrecision. Also obtaining high scores were the relations Mneg and Aexs. The more problematic relations were Com and Voc and modifiers such as Madv, Mquant and Mpre — which can sometimes be ambiguous even for human annotators. They all obtained lower scores. Amongst the modifiers the labeled scores of Mpre were especially low, due to the confusion between it and Anonagr when deciding whether a preposition is an argument or a modifier of a verb, in certain cases a decision that could be hard for a human annotator. Out of the 59 times the gold label was Mpre in the test set of the All-All configuration, 15 times it was labeled as Anonagr by the parser — more than a quarter of the occurrences.

To show a learning curve of the parsers, we trained the parsers on a training set varied in size — starting from 400 utterances up to 3200 utterances with increments of 400. Figure 5.1 shows the learning curves of MEGRASP and MaltParser. The learning curves are expressed by UAS and LAS as a function of the number of utterances the parser is trained on when running the in-domain evaluation scenario. It is important to recall that in the in-domain evaluation scenario the data is ordered chronologically by the age of the target child and the held-out set always succeeds the training set

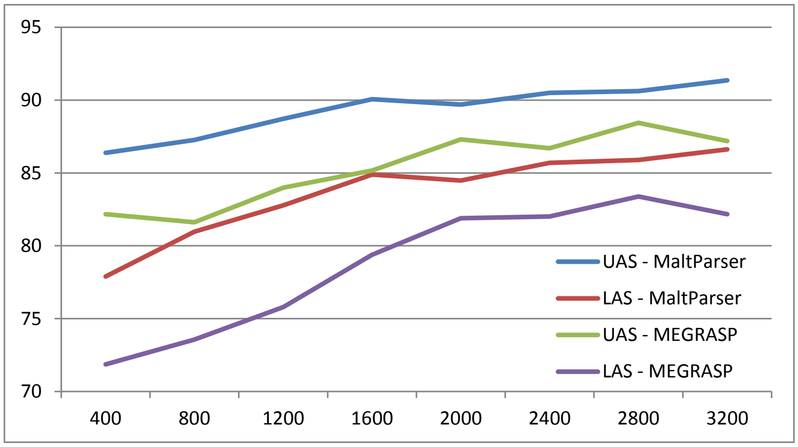

Figure 5.1: MEGRASP and MaltParser in-domain learning curves.

(i.e., of the same child at a same or later age). Training and testing are done on the All-All configuration. The size of the test set is 590 utterances (2474 tokens) for all training sizes. The learning curves of both MaltParser and MEGRASP suggest that more training data can further improve the accuracy of the parser.

5.2 Out-of-domain Evaluation

We now evaluate the parsers on a different domain than the one they were trained on. We trained the parsers on the 8 files of the Ravid corpus. We then parsed the 4 files of the Berman corpus. Table 5.4 shows scores of all data types for MEGRASP and MaltParser.

				MEGRASP			MaltParser		
Train	Size	Test	Size	UAS	LAS	EXM	UAS	LAS	EXM
All	4107	All	1614	78.4	73.1	51.3	82.0	77.1	55.6
CS	1541	CS	761	69.2	61.4	42.0	74.7	68.0	50.7
CDS	2566	CDS	853	81.3	76.3	48.8	85.0	79.7	53.6
CDS	2566	CS	761	73.6	66.6	47.7	77.8	72.1	55.5

Table 5.4: Results: Accuracy of parsing, out-of-domain.

The results confirm the expected differences between the four configurations as discussed in section 4.3. Accuracy in the out-of-domain evaluation scenario is considerably lower than in the in-domain evaluation scenario — up to 22.7 UAS and 25.9 LAS percentage points by MEGRASP and 19.2

UAS and 21.1 LAS percentage points by MaltParser in the CS-CS configuration. The decrease in accuracy when parsing the CS data type can be explained by the fact that the test set of the Berman corpus contains utterances by four different children, all different from the child who is recorded in the training set. They are also children of different ages where three of the four children in the test set are recorded at an older age than the child in the training set. There are some relations that are harder for the parsers to parse that were more common in the CS test domain than in the CS training domain. For example, the problematic `Coord` and `Mpre` relations were more frequent in the test domain (relative to the size of the training set and the test set).

The smallest difference was witnessed by MEGRASP and MaltParser when they were trained and tested on CDS — a decrease of only 4.1 UAS and 4.5 LAS percentage points for MEGRASP and 4.2 UAS and 4.2 LAS percentage points for MaltParser. This should perhaps be contributed to the smaller variance that is expected in CDS between different adults in contrast to the relatviely substantial differences in CS. The Berman corpus also contains on average longer utterances (as can be seen in Table 4.1) which are harder to parse and in addition contribute to an even larger decrease in EXM scores.

Another point to notice is that MaltParser performs better than MEGRASP in this scenario but the differences between the parsers are slightly smaller in some metrics than in the in-domain evaluation scenario. One possible explanation is that MaltParser is run with optimized parameters as suggested by MaltOptimizer (e.g., parsing algorithm and feature set) that are configured according to the training set. In the out-of-domain evaluation scenario the differences in the types of utterances between the training set and the test set are more substantial than in the in-domain evaluation scenario. As a result the optimized parameters are less effective and hence the accuracy is poorer. Still, the advantage of MaltParser over MEGRASP in the All-All configuration is significant for all three metrics ($p < 0.05$).

From these results it appears that when parsing child speech it is better to learn from child-directed speech than from child speech. The UAS, LAS and EXM in the CDS-CS configuration are greater than in the CS-CS configuration — more substantially for MEGRASP. One clear advantage for the CDS-CS configuration is that it has a bigger training set. To further examine this result we trained the parsers on a CDS dataset which is similar in size to the CS dataset (i.e., the training set consists of 1541 CDS utterances). Table 5.5 shows the results of the modified CDS-CS evaluation (line 2) compared to the CS-CS evaluation (line 1) and the original CDS-CS evaluation (line 3).

				MEGRASP			MaltParser		
Train	Size	Test	Size	UAS	LAS	EXM	UAS	LAS	EXM
CS	1541	CS	761	69.2	61.4	42.0	74.7	68.0	50.7
CDS	1541	CS	761	72.8	64.9	45.3	76.4	69.8	49.7
CDS	2566	CS	761	73.6	66.6	47.7	77.8	72.1	55.5

Table 5.5: Results: Accuracy of parsing, out-of-domain, CDS vs. CS.

MEGRASP achieved an UAS of 72.8, a LAS of 64.9 and an EXM of 45.3, still higher than in the CS-CS configuration though as expected lower than the original CDS-CS. MaltParser achieved an UAS of 76.4, a LAS of 69.8 and an EXM of 49.7, the first two are higher than in the CS-CS configuration. The difference in UAS and LAS between the CDS-CS and CS-CS configurations when the training size is the same is statistically significant for MEGRASP ($p < 0.05$) but not quite for MaltParser. So it seems that there is some advantage to training on child-directed speech when parsing child speech. This is in contrast to the trend that was evident from the in-domain task that it is better to train on child speech.

As in the in-domain evaluation scenario, we present a learning curve of the parsers when parsing the same out-of-domain dataset on training sets varying in size. Figure 5.2 shows the learning curves of MEGRASP and MaltParser in the out-of-domain evaluation scenario. The size of the test set is 1614 utterances (8750 tokens). Here, too, the learning curves of both parsers suggest that there is room for improvement with more training data.

5.3 Cross-validation

In addition to evaluating our annotation scheme on the same domain and on a different domain, we want to test it on the corpora as a whole without any distinction to participants or ages. To this end we evaluate the entire set of 12 files (concatenated into one large file) using 5-fold cross-validation. The results are presented in Table 5.6.

		MEGRASP			MaltParser		
Train	Test	UAS	LAS	EXM	UAS	LAS	EXM
All	All	84.0	79.3	60.6	89.5	85.8	70.2

Table 5.6: Results: 5-fold cross-validation.

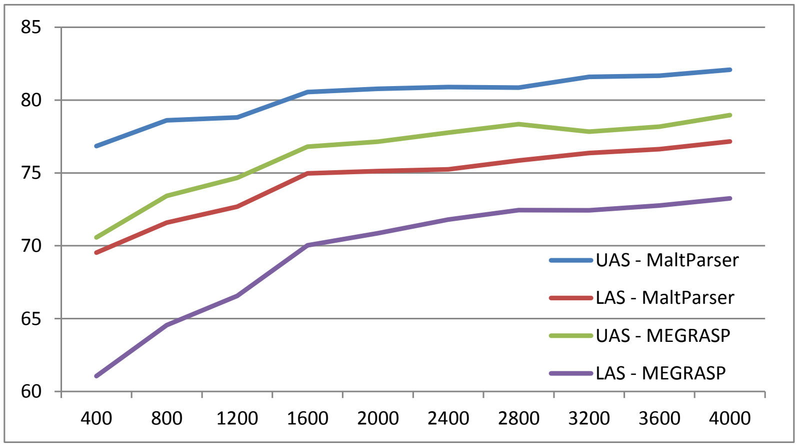

Figure 5.2: MEGRASP and MaltParser out-of-domain learning curves.

These results show the advantage of MaltParser when the entire data is evaluated together. It also shows the robustness of both parsers using the annotation scheme regardless of the domain.

In addition, we performed a similar 5-fold cross-validation on the 8 files of the Ravid corpus. In this scenario the training set and the test set of each fold is of the same domain. Table 5.7 presents these results.

		MEGRASP			MaltParser		
Train	Test	UAS	LAS	EXM	UAS	LAS	EXM
All	All	87.0	82.2	63.5	90.8	86.7	71.0

Table 5.7: Results: 5-fold cross-validation, Ravid corpus.

It is interesting to compare the accuracy of MaltParser in this scenario to the results of the evaluation of CHILDES in English (Sagae et al., 2010). Cross-validation on the Eve corpus of the English section of CHILDES (using MEGRASP) yielded an average result of 93.8 UAS and 92.0 LAS. However, the training set was considerably larger — about 60,000 tokens compared to about 15,000[2] in the training set of each fold of our in-domain cross-

[2] Note that this differs from the size of the Ravid corpus presented in table 4.1, since here we include all tokens (including punctuation) where in the table we do not.

validation evaluation.

5.4 Feature Improvement

As mentioned in section 4.2, we would like to improve the accuracy of certain relations that perform poorly on the default feature configuration. We do this by modifying the feature set that the classifiers use to correctly predict the next attachment or relation.

Currently the feature set is modifiable only in MaltParser. The features used by MEGRASP, that contribute to the maximum entropy classifier during training, are not. That is why MaltParser is used in this section. The features used by the classifiers of MaltParser may vary according to the exact type of the parsing algorithm selected. Recall that MaltParser supports a number of parsing algorithms, which make use of different data structures and different feature sets.

In these evaluation scenarios MaltParser can utilize the following morphological information of each token: gender, number, person and form. Not all tokens contain information regarding these four features, but in the ones that do they may be helpful. In our context, the gender, number and person of a token are relevant for determining agreement. The argument of a verb can be either an agreeing argument (specified by the `Aagr` relation) or a non-agreeing argument (specified by the `Anonagr` relation). In most cases the agreement between the verb and its argument is in gender, number and person.

The 'form' feature of a token can indicate whether a verb is in the imperative mood or the infinitive mood. Finite verbs do not contain form information. More specifically, the 'form' symbol can help determine the `Ainf` relation. The `Ainf` relation is headed by a noun or a verb whose dependent is a verbal element in the infinitive form. This constraint on the relation is rather strict as a token cannot be attached by the `Ainf` relation unless it is marked with the infinitive form in the mor tier.

In some cases the optimized parameters for MaltParser suggested by MaltOptimizer, as used in the in-domain and out-of-domain evaluation scenarios, already include morphological features. In this section we examine their impact. We start with a feature set that does not include any of the four morphological features. We then add different subsets of features to the feature set and evaluate the accuracy of MaltParser using these features. We refer to the MaltOptimizer set of features that does not make use of morphological features as *NoMorph*.

Recall that most of the algorithms that MaltParser supports use two data structures: a *stack* that holds tokens that have already been processed and a *queue* that holds tokens yet to be processed. The features reflect positions within these data structures where '0' indicates the first position. For example, the feature 'number of Stack [0]' specifies the morphological feature 'number' of the token in the first position (i.e., the top) of the stack data structure.

Table 5.8 shows the accuracy of parsing with different features added to the NoMorph feature set. We evaluated MaltParser on the in-domain task with the All-All configuration. The test set is 590 utterances (2474 tokens). Different feature sets containing morphological information improve the accuracy of MaltParser compared to the feature set that does not include morphological information. Although 'form' provided some improvement in itself (line 2), it did not provide further improvement when added to the combination of 'person', 'number' and 'gender' (line 3). The accuracy of parsing improved by 1.1 LAS and 0.9 UAS percentage points. These improvements, however, are not statistically significant ($p > 0.1$).

MaltParser in-domain					
Train	Test	Feature set	UAS	LAS	EXM
All	All	NoMorph	90.5	85.9	71.7
All	All	NoMorph + 'form' (Queue [0])	90.9	86.2	71.9
All	All	NoMorph + 'person', 'number' and 'gender' (Stack [0], Stack[1] and Stack [2])	91.6	86.8	71.5

Table 5.8: Results: Feature improvements. *NoMorph* is the MaltOptimizer suggested feature set without the morphological extended information.

We now show the improvements in specific relations after using the feature set that helped obtain the best accuracy. Table 5.9 shows the changes in the scores of individual relations when adding 'person', 'number' and 'gender' (Stack [0], Stack[1] and Stack [2]) compared to the NoMorph features. Note that the number of occurrences of the relation `Root` is not identical to the number of utterances since in some cases a complex relation (e.g., `Aagr-Root`) was used instead (see section 3.2.5).

This set of features improved the scores of these relations (except `Anonagr`) in almost every metric. Note also that for some relations the increase in the labeled scores is higher than in the unlabeled scores, indicating the contribution of the features to identifying the grammatical relation correctly.

		UAS		LAS	
Relation	Occurrences	Recall	Precision	Recall	Precision
Root	585	+0.8	+1.1	+0.8	+0.9
Anonagr	295	+0.7	-0.8	-0.4	-1.7
Aagr	343	+1.5	+1.0	+2.9	+0.8
Ainf	16	+6.2	+10.0	+12.5	+14.9

Table 5.9: Results: Feature improvements, individual relations. Occurrences refers to the actual number of times this relation appears in the test set.

It is worth mentioning the results presented in Marton et al. (2013) in which the effects of morphological features were extensively evaluated in written Arabic using MaltParser. The characteristics of their work are somewhat different than ours due to the fact that written Arabic is evaluated, which introduces different complexities, and also due to the different experimental setting. But despite the fact that the works are not comparable, it is interesting to discuss their results.

Recall that Arabic is a Semitic language that shares many characteristics with Hebrew. For parsing Arabic they show that the most important features when using gold POS tags and feature values are ‘case’ and ‘state’. The former is not relevant in Hebrew whereas the latter is also relevant to Hebrew (for identifying nouns in the construct state) but was not used in our work since construct state nouns were not common. ‘Gender’, ‘number’ and ‘person’ did not seem to further improve accuracy of parsing Arabic when using gold morphology (significance was not presented when gold morphology was used). However, ‘gender’, ‘number’ and ‘person’ proved to be more helpful when using non-gold POS tags and feature values, probably due to their high prediction rate (Marton et al., 2013).

5.5 Conclusions

The evaluation described in this chapter outlines some interesting results. First, MaltParser proves to be significantly better than MEGRASP in the in-domain and the out-of-domain scenarios. This advantage is probably due to the parameter optimization we ran before each experiment.

Moreover, it seems that when parsing a child-speech corpus different than the one trained on, it is better to train on child-directed speech than child-speech. Also, child-speech suffered the biggest decline in accuracy compared to the scenario where the training corpus and the test corpus are identical, as opposed to child-directed speech for which the decline was minimal. These results meet our expectations that the variance in language between children of different ages, is greater than between adults.

Furthermore, morphological information such as gender, person and number contribute to parsing accuracy, probably due to the contribution of these features for identifying agreement (e.g., between the main verb and the subject).

After evaluating the annotation scheme thoroughly, in the next chapter we investigate alternatives to the scheme for certain constructions which have more than one linguistically plausible analysis, as well as an alternative for token representation.

Chapter 6

Linguistic Issues

Sometimes, a syntactic construction can be annotated in two different ways, and both ways are considered valid by various linguistic theories. This section discusses some linguistic phenomena that may have more than one plausible syntactic annotation. Although the parsers used in this work are based on the notions of Dependency Grammar, we remain agnostic as to the theoretical arguments for either analysis. Instead, we take an empirical approach that examines which annotation might prove most beneficial for the parser. If a particular option turns out to be significant for syntactic parsing, this could be discussed in light of possible theoretical explanations. We annotated the corpora with the different approaches to the phenomena and checked empirically which is more accurate.

The following sections describe four linguistic phenomena that have multiple plausible representations — the first three consider different ways to define relations in the annotation scheme, and the fourth considers an alternative token representation. For each phenomenon we list the accuracy of parsing for the competing representations. The evaluations were conducted on the All-All configuration in the in-domain task, thus the size of the training set is 3286 utterances (11155 tokens) and the size of the test set is 590 utterances (2474 tokens). The All-All configuration was chosen since it has a larger training and test sets and since we wanted to examine these phenomena on the entire data and not separately on CS and CDS. Since MaltParser proved to be the better parser in this domain, only MaltParser was used in this section. Whenever a change in the scheme was made when testing for annotation approaches, the MaltOptimizer process we re-run in order to obtain the optimized parameters for the adjusted annotation.

It is important to note that cross-experiment evaluation is a problematic endeavor, as in the case of comparing between annotation schemes of the same parser. Traditional scoring methods, such as UAS or LAS, are sensitive to annotation choices, and thus differences in accuracy results do not necessarily reflect a true difference in performance (Tsarfaty et al., 2011). Regarding the experiments that are described in this chapter, some of them are based on splits that were made only for training purposes, after which the splits are reversed, and so there seem to be no representation-related issues. However, in other cases the results we present do not necessarily reflect true performance differences between schemes and have to be taken with a grain of salt.

Recently, a platform for cross-experiment evaluation called *TedEval* (Tsarfaty et al., 2011) was developed, allowing to convert syntactic trees based on different annotation schemes and compare annotation schemes or parsers without issues that are related to representation decisions. However, to this date TedEval does not support costume formats (such as the format we used in this work) and so we did not use it in our evaluation.

6.1 Copula Constructions and Other Forms of *hayā* "be"

The first type of utterance that we examined is utterances with some form of the verb *hayā* "be". In Hebrew, the verb *hayā* "be" can be expressed in a variety of constructions, including as an existential marker and a copula (Rosen, 1966). In the present tense *hayā* is in some cases optional and can take the form of a pronoun (in case it expresses a copula) or the words *yeš* "there_is" or *ʔeyn* "there_is_not" (existential markers). In the following sections we refer to this family of tokens (the inflections of *hayā* and the present tense forms) as *hayā form* and to the constructions containing it as *hayā constructions.*

6.1.1 Related Work

Various approaches for copula constructions were suggested in the context of dependency grammar. The original English scheme of CHILDES (Sagae et al., 2010) refers to the verbs 'be', 'become', 'get' etc. as the heads and to the nominal predicates as the dependents, as exhibited in Example 6.1.

(6.1) *Mary* *is* *a* *student*
1|2 2|0 3|4 4|2

Sagae et al. (2010) refers to these verbs as a sub-class of general verbs in the sense that they are the root of the sentence, and the difference is that they select a nominal or adjectival predicate instead of an object. This analysis is consistent with verbal sentences since there is no ellipsis of copular elements in English, and similarly to verbs they are the elements carrying the tense marking. In Hebrew, however, the copula is optional and so relying on the English scheme of CHILDES does not seem to be the best alternative.

The Stanford Parser English scheme (de Marneffe and Manning, 2008a) took a different approach — it refers to the nominal predicate as the head and the copula as the dependent. The subject is also dependent on the nominal predicate. According to de Marneffe and Manning (2008b), the motivation for this decision was to create a scheme which is adaptable to other languages in which there is not necessarily a representation of the copula, and also to help applications extract the semantic information of the sentence through the direct relation between the subject and the predicate. The following in an example from de Marneffe and Manning (2008a):

(6.2) *Bill* *is* *big*
1|3 2|3 3|0

In the Prague Czech Dependency Treebank (Hajičová et al., 1999), the copula is the head of other parts in the sentence. Specifically, the nominal predicate is dependent on the copula. The copula can appear in the past tense and in the future tense (as well as the present), similarly to Hebrew. However, only the Czech verb equivalent of 'be' (být) is regarded as a copula, while Czech equivalents of 'become' and others, that are considered as copular elements in other Czech grammars, are not considered as such (Hajičová et al., 1999). The following is an example from Hajičová et al. (1999):[1]

(6.3) *Pivo "beer"* *je "is"* *zdravé "healthy"*
1|2 2|0 3|2
"Beer is healthy."

When there is zero copula, usually the parts that would depend on the copula (had it existed) are dependent on the token that the copula would depend on. In the Czech scheme, a special relation called ExD is given to

[1] The syntactic analyses of the examples in this section do not necessarily label the dependencies, either because the original work did not label them or because the label names are not relevant in our context.

mark the ellipsis. This is a relation that indicates that the dependency is not direct as it fills the void of an elided item. For example, consider the following sentence (also from Hajičová et al. (1999)):

(6.4) *šatna “cloak-room”* *naproti “opposite”*
`1|0|ExD` `2|0|ExD`
“The cloak-room is opposite.”

In the Prague Arabic Dependency Treebank (Hajič and Zemánek, 2004), the group of verbs referred to as “kana (‘be’) and her sisters” which may act as a copula are regarded as a subset of the entire verb group and thus are the heads of their construction.

Sentences with zero copula are analyzed by placing the nominal predicate as the head — as opposed to marking an ellipsis as was done for Czech. The following is an example from Hajič and Zemánek (2004):

(6.5) *al-’amru “The-matter”* *wāḍiḥun “clear”*
`1|2` `2|0`
“The matter is clear.”

In sentences with pronouns filling a grammatical role of eliminating the determination, the pronoun is dependent on the nominal predicate. The following is an example taken also from Hajič and Zemánek (2004):

(6.6) *ar-rajulu “The-man”* *huwa “he”* *al-muhandisu “the-engineer”*
`1|3` `2|3` `3|0`
“The man is the engineer.”

The approach for Arabic is quite different from the approach for Czech despite the fact that the dependency relations of the Arabic Treebank are based on the Czech dependency relations.

The approach in the Hebrew annotation scheme of Goldberg (2011) is similar to the ideas brought up in the Prague Arabic dependency scheme, in the sense that it first treats the case where there is zero copula and based on that it presents the case where the copula is present.

In a nominal sentence in the present tense with zero copula, the nominal predicate is the head and the subject is its dependent:

(6.7) *ha- “the”* *yēled “child”* *xaḳām “smart”*
`1|2` `2|3` `3|0`
“The child is smart.”

In a nominal sentence in the present tense with a copula, the nominal predicate is also the head and the copula is dependent on it:

(6.8) *ha- “the”* *yēled “child”* *huʔ “he”* *xaḳām “smart”*
`1|2` `2|4` `3|4` `4|0`
“The child is smart.”

The past and future forms of *hayā* are considered as an auxiliary and not a copula. They are the root of the sentence they appear in:

(6.9) *ha- “the”* *menorā “lamp”* *haytā “be”* *semel “symbol”*
`1|2` `2|3` `3|0` `4|3`
xašūv “important”
`5|4`
“The lamp was an important symbol.”

6.1.2 Copula Constructions in Hebrew

In Hebrew, the copula is optional in nominal sentences that are treated as present tense (Berman, 1978), and zero copula constructions are rather common. The same sentence can appear without a copula, such as in *ʔeitan gavōah* (‘Eitan tall’); or with a copula in the form of a pronoun, such as in *ʔeitan huʔ gavōah* (‘Eitan he tall’). The former sentence is analyzed with *gavōah* “tall” marked as the head of the sentence and *ʔeitan* “Eitan” its dependent.

The question is whether to consider the copula (e.g., *huʔ* “he” in the sentence *ʔeitan huʔ gavōah* (‘Eitan he tall’)) as the head of the ensuing nominal or adjectival predicate, or to mark the nominal or adjectival predicate as the head on which both the subject and the copula are dependent. On the one hand, the copula *huʔ* “he” carries the tense marking of the sentence so marking it as the head is consistent with the form of the sentence when it is inflected in the past, *ʔeitan hayā gavōah* (‘Eitan was tall’), in which *hayā* “was” carries the tense marking as well and cannot be omitted. On the other hand, marking *gavōah* “tall” as the head is consistent with the zero copula version of this utterance in which *gavōah* “tall” is also marked as the head.

In order to evaluate the possible contribution of one consistent analysis over the other, we annotate copula utterances — and likewise, other utterances with an *hayā* form — in two ways: one marking the copula as the head and the other marking the nominal predicate as the head. Our goal is to determine which syntactic choice yields better performance and represents the data more adequately.

6.1.3 Conversion Methodology

We describe here the methodology that was implemented to convert the corpora from the annotation of one linguistic approach to the other. We used the label `Acop` to mark the relation between the copula and its argument in which the copula is the head (approach A). We used the label `Xcop` to mark the relation in which the copula is the dependent (approach B). Similarly, we used the labels `Aexs` and `Xexs` for the two approaches of the existential marker. The process that converts one annotation approach to another searches for all occurrences of the grammatical relations `Acop` and `Aexs`, reverses the direction of the attachment between the *hayā* form and the nominal predicate and changes the name of the relations. Following this change other changes are needed — for example, the head of the `Aagr` relation is changed from the *hayā* form to the nominal predicate. Thus the two tokens that were arguments headed by the copula are now directly attached to one another. Furthermore, if the root of the utterance has changed as a result then the heads of the tokens that are dependent on the root are changed as well.

Example 6.10 presents an utterance containing an existential marker annotated according to approach A, where the head is the existential element.

(6.10)	*yeš "there_is"*	*le "to"*	*hi? "she"*	*dimyōn "imagination"*
	`exs`	`prep`	`pro:person`	`n`
	`1|0|Root`	`2|1|Anonagr`	`3|2|Aprep`	`4|1|Aexs`

"She is imaginative."

Example 6.11 presents the same utterance after the conversion to approach B, where the head is the nominal element.

(6.11)	*yeš "there_is"*	*le "to"*	*hi? "she"*	*dimyōn "imagination"*
	`exs`	`prep`	`pro:person`	`n`
	`1|4|Xexs`	`2|1|Anonagr`	`3|2|Aprep`	`4|0|Root`

"She has imaginative."

6.1.4 Results

We present the evaluation of the two linguistic approaches. We trained MaltParser on data annotated with both approaches and evaluated the accuracy of parsing by comparing it to the gold standard test set. This was done for the in-domain evaluation scenario in the All-All configuration. The test set contained 590 utterances (2474 tokens) out of which 45 utterances (271 tokens) contained at least one occurrence of either the relation `Acop` (`Xcop` in approach B) or the relation `Aexs` (`Xexs` in approach B) according to the gold standard annotation. Table 6.1 shows the accuracy of parsing with the two alternatives of *hayā* constructions.

			Approach A		Approach B	
Train	Test	Parser	UAS	LAS	UAS	LAS
All	All	MaltParser	91.2	86.6	90.9	86.3

Table 6.1: Linguistic issues: *hayā* constructions.

Marking the predicate of *hayā* constructions the head instead of the *hayā* form itself results in a decrease of 0.3 UAS percentage points and of 0.3 LAS percentage points. The differences for UAS and LAS between the two approaches are not statistically significant ($p > 0.1$).

Table 6.1 shows accuracy of parsing the entire data set. We also present results for the utterances directly affected by the conversion process, i.e. utterances that contain at least one occurrence of `Acop` (`Xcop`) or `Aexs` (`Xexs`). Table 6.2 shows the accuracy when evaluating the alternative approaches only on the 45 utterances (271 tokens) that contain the `Acop` (`Xcop`) or `Aexs` (`Xexs`) relations.

			Approach A		Approach B	
Train	Test	Parser	UAS	LAS	UAS	LAS
All	All	MaltParser	90.0	88.6	87.4	85.2

Table 6.2: Linguistic issues: Only utterances containing *hayā* constructions.

On this set of utterances there seems to be a decrease in unlabeled and labeled accuracy when switching from approach A to approach B, though as in the general set of utterances these differences are not statistically significant ($p > 0.1$). So overall it seems that there is a slight advantage for approach A.

As discussed in section 6.1, copula-less constructions are rather common in Hebrew in the present tense. Although copula-less utterances are much more common than utterances with an *hayā* form in the present tense (with the agreeing argument present in the utterance) in the training set of the in-domain evaluation scenario, there are only around 50 of them — just above 1 percent of all utterances. Nominal predicates are more frequently dependent on verbs and thus marking them as the root of utterances containing an *hayā* form creates inconsistency.

6.2 Accusative Marker ʔet

We now proceed to the second linguistic issue we examined, the accusative marker *ʔet*. The distribution of the accusative marker ʔet is rather unique. On the one hand, it behaves much like a preposition: it can either introduce a lexical noun phrase or inflect with a pronominal suffix, and it expresses Verb-Patient. On the other hand, it appears only before definite direct objects, while there is no accusative case marking for indefinite complements or in compound constructions with abstract nominalizations as heads. (Berman, 1978). There are a few other languages that exhibit this phenomenon, for example Persian (Ganjavi, 2007). Many other languages also mark nouns for case (either by inflection or by independent morphemes) without making a distinction between definite and indefinite nouns. The ʔet accusative marker can be viewed as part of the preposition family (thus heading its noun complement) or as a vacuous morpheme (thus dependent on its complement) (Danon, 2006). So a decision is to be made with regards to the identity of the head of a definite accusative complement — whether it is the ʔet accusative marker, or the noun itself.

1. One can view ʔet-phrases as prepositional phrases and annotate them accordingly: The accusative marker is marked as an argument of the verb, and the nominal element is the argument of ʔet. We label the relation between the verb and the accusative marker `Anonagr` and between the accusative marker and the nominal element `Aprep`. In the following example, *rocē* is the head and *ʔet* is its dependent. Also, *ʔet* is the head and *ṭipōt* is its dependent, being the argument of a preposition (in this case, the type of the preposition is an accusative marker):

(6.12)	*loʔ* *"no"*	**rocē** *"want"*	**ʔet** *"AT"*	*ha-* *"the"*
	`neg`	`part|num:sg`	`acc`	`det`
	`1|2|Mneg`	`2|0|Root`	`3|2|Anonagr`	`4|5|Mdet`

ṭipōt *"drops"*
`n|num:pl`
`5|3|Aprep`
"(I) don't want the drops!"

2. Alternatively, the nominal element can be viewed as directly dependent on the verb, with the ʔet accusative marker being vacuous and not part of the preposition family. The nominal element is the argument of the verb with a relation labeled `Anonagr`, and the accusative marker depends on the nominal element with a relation labeled `Xacc`. In the following example, *rocē* is the head and *ṭipōt* is its dependent. Also, *ṭipōt* is the head and *ʔet* its dependent:

(6.13)					
	lo ʔ “no”	**rocē** *“want”*	**ʔet** *“AT”*	*ha- “the”*	**ṭipōt** *“drops”*
	`neg`	`part|num:sg`	`acc`	`det`	`n|num:pl`
	`1|2|Mneg`	`2|0|Root`	`3|5|Xacc`	`4|5|Mdet`	`5|2|Anonagr`

“(I) don’t want the drops!”

The first analysis has an advantage of being consistent with the analysis of prepositional phrases in general, with *ʔet* being the head of the complement phrase just like prepositions are. However, when the complement is indefinite, no *ʔet* appears, like in the following utterance where *rocē* is the head and *ṭipōt* is its dependent:

(6.14)			
	loʔ “no”	**rocē** *“want”*	**ṭipōt** *“drops”*
	`neg`	`part|num:sg`	`n|num:pl`
	`1|2|Mneg`	`2|0|Root`	`3|2|Anonagr`

“(I) don’t want drops!”

Considering Example 6.14, the second analysis presented above has an advantage as it is consistent with indefinite constructions with no *ʔet*, as both mark the nominal element (e.g., *ṭipōt*) as the direct dependent of the verb (e.g., *rocē*).

In the Hebrew annotation scheme of Goldberg (2011), the marker ʔet is the head of the nominal element. Goldberg (2011) states that the reason for this decision is to adapt to cases where the marker ʔet may appear whereas the subsequent nominal element is elided (e.g., *šamāʕti ʔet še ʔamārta* (‘I-hear that ACC you-say’)). These types of sentences are rather formal and we did not encounter them in spoken language.

Since the accusative marker ʔet is an interesting and unique problem we evaluated the different annotations for these constructions and checked on which the parser performs better. We annotate our corpora according to both approaches and see which approach yields better accuracy.

6.2.1 Conversion Methodology

We now describe the method implemented to convert the corpora from the annotation of one linguistic approach to the other. The process searches occurrences of the ʔet token analyzed morphologically as *acc* (accusative). The direction of the attachment between these tokens and their nominal dependents is reversed and the labels of the relation are modified. Since these constructions are dependent on a verb the head of the nominal element should be changed to be that verb. Few side-effects apply as a result of this conversion — the main one is when the accusative marker was previously the root of the utterance. In this rare case the tokens that were dependent on the accusative marker are changed to depend on the nominal element which is the new root.

Example 6.15 presents an utterance containing an accusative marker annotated according to approach A, where the head is the accusative marker.

(6.15)

Sivān "Sivan"	*tinʕāl "shoe"*	*ʔet "ACC"*	*ha "the"*	*nāʕal "shoe"*
n:prop	v	acc	det	n
1|2|Aagr	2|0|Root	3|2|Anonagr	4|5|Mdet	5|3|Aprep

"Sivan will put on her shoe."

Example 6.16 presents the same utterance after the conversion to the annotation of approach B, where the head is the nominal element.

(6.16)

Sivān "Sivan"	*tinʕāl "shoe"*	*ʔet "ACC"*	*ha "the"*	*nāʕal "shoe"*
n:prop	v	acc	det	n
1|2|Aagr	2|0|Root	3|5|Xacc	4|5|Mdet	5|2|Anonagr

"Sivan will put on her shoe."

6.2.2 Results

We present the evaluation of the two linguistic approaches. We trained MaltParser on data annotated in accordance with both approaches and evaluated the accuracy of parsing by comparing it to the gold standard test set. This was done for the in-domain evaluation scenario in the All-All configuration.

Table 6.3 shows the accuracy of parsing for the two alternatives. The test set contained 590 utterances (2474 tokens) out of which 41 utterances (215 tokens) contained at least one occurrence of an accusative marker.

			Approach A		Approach B	
Train	Test	Parser	UAS	LAS	UAS	LAS
All	All	MaltParser	91.2	86.6	90.6	86.1

Table 6.3: Linguistic issues: Accusative marker.

When annotating the accusative marker as the head there is an advantage of 0.6 UAS percentage points and of 0.5 LAS percentage points. The differences between the two approaches are not statistically significant ($p > 0.1$).

We also show the accuracy when parsing only utterances that contain an accusative marker. Table 6.4 shows the accuracy when evaluating the alternative approaches only on the 41 utterances (215 tokens) containing an accusative marker.

			Approach A		Approach B	
Train	Test	Parser	UAS	LAS	UAS	LAS
All	All	MaltParser	95.3	92.6	94.9	90.7

Table 6.4: Linguistic issues: Only utterances containing an accusative marker.

On this set of utterances there also seems to be an advantage to approach A, though the number of utterances that this change effected is relatively small. The differences are not statistically significant ($p > 0.1$). Recall that the two alternatives we consider regarding the accusative marker are whether it acts as a preposition or a vacuous case marker. The fact that the results have not increased when switching from the annoation of approach A to approach B could be explained by frequency analysis of the `Anonagr` relation — the `Anonagr` relation between a verb and a preposition (supporting the preposition approach) and between a verb and a noun (supporting the vacuous marker approach) are both relatively common in the training data.

We also annotated the corpora with approach B of accusative marker constructions and approach B of *hayā* constructions. This combined approach performed slightly worse than the original combined approach A as the UAS is 90.8 and the LAS is 86.0.

6.3 Prepositional Arguments of Verbs

We attempted to improve the accuracy of parsing when dealing with constructions that include prepositional arguments of verbs. A prepositional phrase following a verb can be the verb's argument or its modifier. There are 1155 cases of a preposition following a verb or a participle in the entire corpora. The decision whether these prepositional phrases are indeed arguments or modifiers of a certain verb is hard even for human annotators.

In Example 6.17, the preposition *le-* "to" that is dependent on the verb *ʔaʕaṣē* "will-do" is considered to be a modifier with the `Mpre` relation (see Appendix A).

(6.17)	*ʔanī "I"*	*ʔaʕaṣē "will-do"*	*le- "to"*
	`pro:person|pers:1&num:sg`	`v|pers:1&num:sg`	`prep`
	`1|2|Aagr`	`2|0|Root`	`3|2|Mpre`

ʔatā "you"	*baʕayōt "problems"*
`pro:person`	`n`
`4|3|Aprep`	`5|2|Anonagr`

"I will cause you problems."

However, in Example 6.18 the preposition *le-* "to" that is dependent on the verb *tagīdi* "say" is considered to be its argument and is thus annotated with the `Anonagr` relation.

(6.18)	*ʕḳšăāyw "now"*	*tagīdi "say"*	*le- "to"*	*ʔanī "I"*
	`adv`	`v`	`prep`	`pro:person`
	`1|2|Madv`	`2|0|Root`	`3|2|Anonagr`	`4|3|Aprep`

Siwān "Sivan"
`n:prop`
`5|2|Voc`

"Now tell me, Sivan."

These subtleties between prepositional arguments and modifiers of verbs lead to poor LRecall and LPrecision of the `Mpre` relation as witnessed in Table 5.3. In order to improve the overall accuracy of the parser we examined a possible solution to this issue. We altered the annotation scheme and created a new relation called `Averb` that uniformly labels the attachment between a verb and a preposition — whether is it an argument or a modifier. The `Mpre` relation remains when a preposition is dependent on a noun, and the `Anonagr` relation now represents arguments of verbs which are not prepositions.

Example 6.19 shows how Example 6.18 is annotated after this change.

(6.19)	*ʕḳšāyw “now”*	*tagīdi “say”*	*le- “to”*	*ʔanī “I”*
	`adv`	`v`	`prep`	`pro:person`
	`1|2|Madv`	`2|0|Root`	`3|2|Averb`	`4|3|Aprep`
	Siwān “Sivan”			
	`n:prop`			
	`5|2|Voc`			

“Now tell me, Sivan.”

6.3.1 Results

To examine the contribution of this change we converted the corpora to reflect the inclusion of the `Averb` relation. We then trained the parsers on the in-domain training set and evaluated the accuracy of parsing by comparing the parsed test set to the gold standard. Table 6.5 shows UAS and LAS after introducing the new `Averb` relation to the scheme (approach B).

			Approach A		Approach B	
Train	Test	Parser	UAS	LAS	UAS	LAS
All	All	MaltParser	91.2	86.6	90.6	86.6

Table 6.5: Linguistic issues: Introduction of the `Averb` relation.

In the All-All configuration, although the accuracy of the `Averb` relation itself is high (over 90 percent in all indvidual metrics) there seems to be a slight overall decrease in unlabeled accuracy. This difference is not statistically significant.

6.4 Token Representation

An interesting topic of discussion is whether the representation of words and their morphological and syntactic analyses should be *word-based* or *morph-based*. A morph-based approach calls for the split of words into morphemes, the atomic units that are combined to create words, whereas a word-based approach refers to words as the minimal units of the language (Blevins, 2006; Tsarfaty and Goldberg, 2008). Recall that in order to reduce sparseness of data, our work involved a pre-processing stage of splitting pronominal suffixes and inflected prepositions to separate tokens, rendering the representation of the corpora partially morph-based. Using our annotated data and a conversion script from one representation to another we can investigate the differences between the word-based approach and the morph-based approach and evaluate these alternatives empirically. More specifically, we examine the accuracy of parsing on data in which pronominal suffixes and inflected prepositions were *not* split. For the representation of the tokens before the split, refer back to section 2.2.2. We compare the performance of the no-split approach to the split approach used throughout this work.

6.4.1 Results

We trained MaltParser on a version of the data that reflects the word-based approach of token representation, namely that no split of pronominal suffixes and inflected prepositions occurs (approach B), and compare the accuracy of parsing to the accuracy obtained by MaltParser on the split data (approach A). This comparison was done in the in-domain evaluation scenario. Table 6.6 shows the accuracy of parsing of the two approaches. There are 574 utterances in the test set (2342 tokens). The tokens that are introduced as a result of the split were not included in the evaluation since the majority of them are labeled `Aprep` on which the parser has a very high accuracy. This way we avoid inflation of the scores of the split approach.

			Approach A		Approach B	
Train	Test	Parser	UAS	LAS	UAS	LAS
All	All	MaltParser	90.7	85.9	90.0	85.1

Table 6.6: Linguistic issues: Token representation.

The results show the advantage of training and parsing using the split version of the data, though the differences are not significant ($p > 0.1$).

To further examine the difference between working with two approaches of token representation we evaluated the two approaches only on utterances containing a split token (e.g., an inflected preposition, including the accusative marker and the possessive marker, or a suffixed pronominal). Since we didn't have a large amount of utterances containing a split token in the test set, we manually annotated around 30 utterances from another file of the Ravid corpus (similarly to the rest of the files we worked with, this file has a gold standard disambiguated morphological tier). We added these manually annotated utterances to the original test set of the in-domain evaluation scenario, thus creating an expanded test set containing 155 utterances (709 tokens) with an inflected preposition or a suffixed pronominal. As before, the tokens introduced as a result of the split are not included in the evaluation. Table 6.7 shows the results of the evaluation of these utterances.

			Approach A		Approach B	
Train	Test	Parser	UAS	LAS	UAS	LAS
All	All	MaltParser	91.3	85.8	90.4	84.9

Table 6.7: Linguistic issues: Token representation, 155 utterances containing an inflected preposition or a suffixed pronoun.

On utterances containing at least one inflected preposition or suffixed pronominal there is an advantage for approach A. The differences in UAS and LAS are not significant ($p > 0.1$). The fact that the advantage for the split data is not significant could partially be contributed to the use of MaltOptimizer separately for each version of data. When the no-split approach is used the optimized parameters are tailored to it and perhaps assist in accuracy of parsing despite the lack of split.

Chapter 7

Conclusions and Future Work

This work presented a new annotation scheme that deals with Hebrew spoken language as part of the Hebrew section of the CHILDES database. The annotation scheme we built handles some of the unique linguistic characteristics of spoken language in general and Hebrew spoken language in particular. We showed that a parser trained on data annotated using our annotation scheme achieves good results when parsing the same domain and also is adaptable to other domains. This is despite a relatively small data set available.

We showed that both MaltParser and MEGRASP produced relatively good results in the in-domain evaluation scenarios using our annotation scheme. In both evaluation scenarios, MaltParser proved to be the better of the two, thanks to parameter tuning done by MaltOptimizer. It is worth noting that the transcriptions at hand were sometimes erroneous or structured problematically. This had an effect on the quality of the syntactic analysis and future acquired data should help in this respect.

We examined the differences between learning from CDS and CS when annotating CS. Within the same domain there was no significant difference and both configurations yielded relatively high accuracy. However, when parsing out of domain there was a clear advantage to training from CDS. We address this probably to the simplicity of the CS in the in-domain scenario as well as to differences in CS between the training set and the test set (and within the test set) in the out-of-domain scenario. We conclude that as expected there is some difficulty adapting CS from one domain to another (also recalling the age gap between the domains) whereas CDS is more stable and less varied between domains.

When we parsed transcriptions of a different domain than the one the parser was trained on, all metrics showed a decrease in accuracy for both parsers. This decrease is expected considering the differences between the domains in terms of MLU, the age of the participants and the frequency of some types of relations. Despite this decrease, the accuracy of parsing in the All-All configuration was reasonable, showing the adaptability of the parsers and the annotation scheme to other domains.

Working with MaltParser allowed us to evaluate the impact of features derived from the morphological tier of the corpora. Although the accuracy of parsing using the feature set without extended morphological data is quite high — attributed to the fact that the basic feature set was optimized by running MaltOptimizer and to the presence of a gold standard morphological tier — when we used detailed morphological information we were able to improve the accuracy of parsing even more. The best accuracy was exhibited using the morphological attributes 'gender', 'person' and 'number'. Future work in this area can embark on a more systematic approach that has a sole purpose of examining the contribution of morphological features. This includes extracting more morphological attributes other than those that were used in this work, as well as a more elaborate search for subsets of features that are derived from MaltParser data structures. Morphological information can also be evaluated on other parsers, namely MEGRASP (though currently MEGRASP does not have an interface for modifying the feature set). The morphological information may have an even bigger impact when using MEGRASP since the default feature set of MEGRASP is rather limited and not tailored specifically to any given training set. This is in contrast with the feature set we used with MaltParser after running MaltOptimizer.

We examined different linguistic approaches for *hayā* constructions and accusative marker constructions. In both cases significant advantages to either approach were not revealed. This is interesting since one of the goals of a linguistic approach in theory is to reflect the phenomenon of the language accurately. The lack of significance could be contributed to the characteristics of this corpora or its size. Another possible explanation to this empirical evaluation may very well be that as long as the annotation is consistent it can produce reasonable results, regardless of what the specific annotation approach we use. We would like to see if this is a cross-linguistic phenomenon, e.g. in copula constructions which are a challenge in several languages.

We utilized the fact that the input to the syntactic process is a fully disambiguated gold standard morphological tier. An interesting extension is to evaluate the parser on data with a morphological tier that was created automatically. Apart from an obvious decrease in accuracy we expect that this may also introduce some different effects when examining feature sets or linguistic issues. Another extension to this work is parsing of Hebrew spoken language from other domains. Using other sources of spoken language necessitates orthographic and morphological processing stages after which the syntactic parser built in this work can be used as a building block to process these sources syntactically.

Appendix A

Taxonomy of Dependents

A.1 Arguments

AgreementArgument [**Aagr**] Identifies the dependent argument with which the predicate agrees. This argument cannot be a clause in itself. Typically, it is a nominal element (noun or pronoun).

In the following example, **mešaqēret** is the head and **ʔat** is its dependent in an `Aagr` relation:

(A.1) **ʔat** *"you"* **mešaqēret** *"lie"*
`pro:person|gen:fm&num:sg` `part|gen:fm&num:sg`
`1|2|Aagr` `2|0|Root`
"You are lying!"

Non-agreementArgument [**Anonagr**] Identifies any argument of a verb with which agreement does not hold. Typically, it is a nominal that is dependent on the verb.

In the following example, **rocē** is the head and **ṭipōt** is the dependent in an `Anonagr` relation. Note that no agreement holds between the two, as the former is in the singular form and the latter is in the plural form:

(A.2) *ʔanī "I"* *loʔ "no"* **rocē** *"want"* **ṭipōt** *"drops"*
`pro:person|num:sg` `neg` `part|num:sg` `n|num:pl`
`1|3|Aagr` `2|3|Mneg` `3|0|Root` `4|3|Anonagr`
"I don't want drops."

In the following example, **ʔohēv** is the head and **meqomōt** is its dependent in an `Anonagr` relation. Note that again no agreement holds between the two, as the former is in the singular and the latter is in the plural form:

(A.3) *ken “yes”* , *“,”* *huʔ “he”*
co , pro:person|gen:ms&num:sg
1|4|Com 2|4|Punct 3|4|Aagr
ʔohēv *“love”* **meqomōt** *“places”* *ʕim “with”*
part|gen:ms&num:sg n prep
4|0|Root 5|4|Anonagr 6|5|Mpre
liḳlūḳ “dirt”
n
7|6|Aprep
“Yes, he loves places with dirt.”

The Anonagr relation can also mark the relation between the main predicate and a non-agreement argument even when it is its lone argument in the utterance.

In the following example, **hīne** is the head and **sapār** is the dependent in an Anonagr relation:

(A.4) **hīne** *“here”* *ha- “the”* **sapār** *“barber”*
co det n
1|0|Root 2|3|Mdet 3|1|Anonagr
“Here is the barber.”

The non-agreement argument may in fact sometimes coincidentally appear to agree with the verb. In the following example, **macāʔ** is the head and **maqōm** is the dependent in an Anonagr relation. Note that the two words share the same gender (masculin) and number (singular), but we still interpret the relation between them as Anonagr based on thematic relations and canonic word order:

(A.5) *hīne “here”* *huʔ “he”* **macāʔ** *“find”*
co pro:person|gen:ms&num:sg v|gen:ms&num:sg
1|3|Com 2|3|Aagr 3|0|Root
maqōm *“place”*
n|gen:ms&num:sg
4|3|Anonagr
“There, he found a place.”

The relation for non-agreeing arguments also applies for what are typically termed *indirect* or *oblique* arguments. In these constructions, the nominal element may be preceded by a preposition. The `Anonagr` dependency is marked on the prepositional element and the nominal element is marked as the argument of a preposition, `Aprep`.

In the following example, **pagāʕti** is the head and **be-** is its dependent in an `Anonagr` relation, being the head of a complement prepositional phrase. Also, **be-** is the head and **ʔat** is its dependent in an `Aprep` relation:

(A.6)	*ʔoy “oh_no”*	*, “,”*	**pagāʕti** *“I-hurt”*	**be-** *“in”*
	`co`	`,`	`v`	`prep`
	`1|3|Com`	`2|3|Punct`	`3|0|Root`	`4|3|Anonagr`

ʔat *“you”*
`pro:person`
`5|4|Aprep`

“Oh no, did I hurt you?”

Note that as stated in section 2.2.2, the fused morpheme *baḳ* “in you” is split into the tokens *be-* “in” and *ʔat* “you”. To show the benefit of such a split, consider the following example, where **pagāʕti** is the head and **be-** is the dependent in an `Anonagr` relation, being the head of a complement prepositional phrase. Also **be-** is the head and **Siwān** is the dependent in an `Aprep` relation:

(A.7)	*ʔoy “oh_no”*	*, “,”*	**pagāʕti** *“I-hurt”*	**be-** *“in”*
	`co`	`,`	`v`	`prep`
	`1|3|Com`	`2|3|Punct`	`3|0|Root`	`4|3|Anonagr`

Siwān *“Siwān”*
`pro:person`
`5|4|Aprep`

“Oh no, did I hurt Sivan?”

Thus, the split allows us to maintain consistency between Examples A.6 and Example A.7 where no split is needed and the complement of the preposition appears in full.

In the following example, **be-** is the head and **bāyit** is its dependent in an `Aprep` relation, being the argument of a preposition:

(A.8)	*yašānti “I-sleep”*	**be-** *“in”*	**ha-** *“the”*	**bāyit** *“house/home”*
	`v`	`prep`	`det`	`n`
	`1|0|Root`	`2|1|Mpre`	`3|4|Mdet`	`4|2|Aprep`

šel “of”	*ʔanī “I”*
`prep`	`pro:person|num:sg`
`5|4|Mposs`	`6|5|Aprep`

“I slept in my home.”

The same dependency relations are noted for cases where the non-agreement argument is preceded by the accusative marker ʔet (typically termed the *direct object*). There are two options to annotate these utterances, both were discussed elaborately in section 6.2: One where the accusative marker ʔet behaves like a preposition, i.e. ʔet is the head and the nominal element is the dependent (marked with an `Aprep` relation); and the other where the accusative marker ʔet is vacuous and is merely a case marker, i.e. the nominal element is the head and the accusative marker ʔet is the dependent (marked with a `Xacc` relation).

In instances where more than one non-agreeing argument occurs in the construction (typically termed ditransitive constructions), the second dependent (typically an indirect object) is marked also as `Anonagr`.

In the following example, **natnā** is the head and **māšehu** is its dependent. Also, **natnā** is the head and **le-** is its dependent in an `Anonagr` relation, being the argument marked with the dative case:

(A.9)

hiʔ “she”	**natnā** *“give”*	**le-** *“to”*
`pro:person|gen:fm&num:sg`	`v|gen:fm&num:sg`	`prep`
`1|2|Aagr`	`2|0|Root`	`3|2|Anonagr`

ʔanī “I”	**māšehu** *“something”*	*be- “in”*	*ha- “the”*
`pro:person`	`pro:indef`	`prep`	`det`
`4|3|Aprep`	`5|2|Anonagr`	`6|2|Mpre`	`7|8|Mdet`

ʔōzen “ear”
`n`
`8|6|Aprep`

“She gave me something in the ear.”

When annotating the second argument of a ditransitive verb one might consider using an `Anonagr2` label (a similar notion was used in Sagae et al. (2010)). One of the reasons we do not term the second argument as `Anonagr2` — differentiating it from the first argument — is that the order of the arguments may well have been reversed. Consider the following example, where **natnā** is the head and **le-** is its dependent in an `Anonagr` relation, being the argument marked with the dative case. Also, **natnā** is the head and **māšehu** its dependent in an `Anonagr` relation:

(A.10)

hiʔ “she”	**natnā** *“give”*
`pro:person|gen:fm&num:sg`	`v|gen:fm&num:sg`
`1|2|Aagr`	`2|0|Root`

<table>
<tr><td><b>māšehu</b> <i>“something”</i></td><td><i>be-</i> <i>“in”</i></td><td><i>ha-</i> <i>“the”</i></td><td><i>ʔōzen</i> <i>“ear”</i></td></tr>
<tr><td>pro:indef</td><td>prep</td><td>det</td><td>n</td></tr>
<tr><td>3|2|Anonagr</td><td>4|3|Mpre</td><td>5|6|Mdet</td><td>6|4|Aprep</td></tr>
</table>

<table>
<tr><td><b>le-</b> <i>“to”</i></td><td><i>ʔanī</i> <i>“I”</i></td></tr>
<tr><td>prep</td><td>pro:person</td></tr>
<tr><td>7|2|Anonagr</td><td>8|7|Aprep</td></tr>
</table>

“She gave something in the ear to me.”

Non-agreement arguments can also occur as finite clausal dependents of the root. As in the case of nominal arguments, we treat the whole construction as depending on the main predicate of the matrix clause. The head of the subordinate clause is the complementizer. It marks a dependency between the main verb in the matrix clause and its non-agreeing clausal argument and it functions as the head on which the main verb of the subordinate clause depends, in a `SubCl` relation.

In the following example, the verb **rocē** in the main clause is the head and the complementizer **še-** its dependent in an `Anonagr` relation. Also, **še-** is the head and the verb **ʔesarēq** its dependent in a `SubCl` relation, being the main verb of a subordinate clause:

(A.11)

<table>
<tr><td><i>ʔatā</i> <i>“you”</i></td><td><b>rocē</b> <i>“want”</i></td></tr>
<tr><td>pro:person|gen:ms&num:sg</td><td>part|gen:ms&num:sg</td></tr>
<tr><td>1|2|Aagr</td><td>2|0|Root</td></tr>
</table>

<table>
<tr><td><b>še-</b> <i>“that”</i></td><td><i>ʔanī</i> <i>“I”</i></td><td><b>ʔesarēq</b> <i>“comb”</i></td></tr>
<tr><td>conj:subor</td><td>pro:person|num:sg</td><td>v|num:sg</td></tr>
<tr><td>3|2|Anonagr</td><td>4|5|Aagr</td><td>5|3|SubCl</td></tr>
</table>

<table>
<tr><td><i>ʔet</i> <i>“ACC”</i></td><td><i>ʔatā</i> <i>“you”</i></td></tr>
<tr><td>acc</td><td>pro:person</td></tr>
<tr><td>6|5|Anonagr</td><td>7|6|Aprep</td></tr>
</table>

“Do you want me to comb your hair?”

NonFiniteArgument [Ainf] This relation is marked between a verb or noun in the main clause and its non-finite verbal argument. It is headed by the verb or noun of the main clause.

In the following example, **titēn** is the head and **laʕavōr** is its dependent in an `Ainf` relation:

(A.12)

<table>
<tr><td><i>ʔaz</i> <i>“so”</i></td><td><b>titēn</b> <i>“you-let”</i></td><td><i>le-</i> <i>“to”</i></td><td><i>ʔanī</i> <i>“I”</i></td></tr>
<tr><td>adv</td><td>v</td><td>prep</td><td>pro:person</td></tr>
<tr><td>1|2|Com</td><td>2|0|Root</td><td>3|2|Anonagr</td><td>4|3|Aprep</td></tr>
</table>

laʕavōr *“pass”*
v
5|2|Ainf
“So let me pass.”

In the case of direct speech complementation, the main predicate of the subordinate clause is connected in an Anonagr relation to the main predicate of the matrix clause.

In the following example, the verb **ʔomēr** is the head and the verb **higīaʕ** is the dependent in an Anonagr relation. Also, **zman** is the head and **laqūm** is the dependent in an Ainf relation:

(A.13)	*huʔ “he”*	**ʔomēr** *“say”*	*le- “to”*	*huʔ “he”*
	pro:person|num:sg	part|num:sg	prep	pro:person
	1|2|Aagr	2|0|Root	3|2|Anonagr	4|3|Aprep

higīaʕ *“arrive”*	*ha- “the”*	*zman “time”*	*laqūm “wake_up”*
v|num:sg	det	n|num:sg	v
5|2|Anonagr	6|7|Mdet	7|5|Aagr	8|7|Ainf

“He tells him: “it’s time to wake up”.”

Ainf also applies to arguments of modals. In the following example, the modal **carīḳ** is the head and **lehistakēl** is its dependent in an Ainf relation:

(A.14)	*hīne “here”*	**carīḳ** *“necessary”*	**lehistakēl** *“look”*	*be- “in”*
	co	adj	v	prep
	1|2|Com	2|0|Root	3|2|Ainf	4|3|Anonagr

ha- “the”	*tmunā “picture”*
det	n
5|6|Mdet	6|4|Aprep

“Here, it is necessary to look at the picture.”

ArgumentOfCopula [Acop] Identifies the relation between a copula and its predicate (either nominal or adjectival), where by copula we consider forms of the verb *hayā* “be” that is transformed into a pronoun in the present tense, and demonstrative pronouns such as *ze* “it” which also appear in the present tense. As discussed in section 6.1, there are typically two options to annotate such utterances: Either the nominal or adjectival predicate is dependent on the copula (marked with an Acop relation), or that the nominal or adjectival predicate head the subject of the utterance and as a result they also head the copula (marked with a Xcop relation).

The first analysis has the advantage of being more consistent with verbal utterances, since as in verbal utterances, the root of the utterance is the element carrying the tense markings of the utterance. However, a problem occurs when the copula is elided, a common phenomenon in Hebrew. In this case the nominal predicate becomes the root of the utterance, like in the following example, where **ʔayēf** is the head and **ʔanī** is its dependent in an `Aagr` relation, being the argument in agreement with the nominal element:

(A.15)	**ʔanī** *"I"*	**ʔayēf** *"tired"*
	`pro:person|num:sg`	`adj|num:sg`
	`1|2|Aagr`	`2|0|Root`

"I am tired."

For cases similar to Example A.15, the second analysis (where the nominal predicate is the head) has an advantage as it is more consistent: regardless of whether the copula is elided or not, the nominal element is the root of the utterance.

For brevity, here are the two possible analyses exemplified by the following utterance:

1. The first option of analysis marks **hayā** as the head and **šam** as its dependent in an `Acop` relation:

(A.16)	*Dani "Dani"*	**hayā** *"be"*	**šam** *"there"*
	`n:prop`	`cop`	`adv`
	`1|2|Aagr`	`2|0|Root`	`3|2|Acop`

"Dani was there."

2. Alternatively, one can view **šam** as the head and **hayā** as its dependent in a `Xcop` relation:

(A.17)	*Dani "Dani"*	**hayā** *"be"*	**šam** *"there"*
	`n:prop`	`cop`	`adv`
	`1|3|Aagr`	`2|3|Xcop`	`3|0|Root`

"Dani was there."

Copula constructions can also involve finite clause predicates. As in the case of arguments dependent on a verb, the main predicate of the subordinate clause is marked as a dependent of the complementizer and the complementizer functions as the argument of the copula.

In the following example, where the syntactic analysis approach regarding copula constructions is the one where the copula element is the head, **hiʔ** is the head and **še-** is its dependent in an `Acop` relation:

(A.18)	*ha- “the”*	*beʕayā “problem”*	**hiʔ** *“she”*	**še-** *“that”*
	`det`	`n`	`pro:person`	`conj:subor`
	`1|2|Mdet`	`2|3|Aagr`	`3|0|Root`	`4|3|Acop`

ʔanī “I”	*ʕacūva “sad”*
`pro:person|num:sg`	`adj|num:sg`
`5|6|Aagr`	`6|4|SubCl`

“The problem is that I’m sad.”

Alternatively, a second analysis is possible, similarly to copula constructions without a finite clause. In this analysis the copula element is dependent on the clause, as in the following example, where **še-** is the head and **hiʔ** is the dependent in a Xcop relation:

(A.19)	*ha- “the”*	*beʕayā “problem”*	**hiʔ** *“she”*	**še-** *“that”*
	`det`	`n`	`pro:person`	`conj:subor`
	`1|2|Mdet`	`2|4|Aagr`	`3|4|Xcop`	`4|0|Root`

ʔanī “I”	*ʕacūva “sad”*
`pro:person|num:sg`	`adj|num:sg`
`5|6|Aagr`	`6|4|SubCl`

“The problem is that I’m sad.”

The past tense form of *hayā* “be” may also refer to an auxiliary. Since auxiliaries were very rare in the corpora, we decided to indicate the relation between auxiliaries and a subsequent participle with `Acop` instead of using a separate relation which the parser would have had a difficult time learning — bearing in mind that this does not mean that an auxiliary is in fact a copula. The construction of an auxiliary and a participle specifies a variety of meanings: past progressive, the conditional mood and polite requests, to name a few.

For this type of construction we also consider two alternative annotations:

1. The relation is headed by the auxiliary. The motivation for this analysis is that the auxiliary is carrying the tense markings of the utterance.

   In the following example, **haytā** is the head and **ʔomēret** is its dependent in an `Acop` relation:

(A.20)

ʔīmaʔ *“mother”*	**haytā** *“be”*
`n`	`cop|gen:fm&num:sg`
`1|2|Aagr`	`2|0|Root`

ʔomēret *“say”*	*le-* *“to”*	*ʔāba*ʔ *“dad”*
`part|gen:fm&num:sg`	`prep`	`n`
`3|2|Acop`	`4|3|Anonagr`	`5|4|Aprep`

“Mom would have said to Dad.”

2. The relation is headed by the participle. The motivation for this analysis is that the auxiliary is optional and appears only in the past tense. In this analysis the auxiliary is considered a verb modifier and the relation is marked with `Xcop`.

   In the following example, **ʔomēret** is the head and **haytā** is its dependent in a Xcop relation:

(A.21)

ʔm̄aʔ *“mother”*	**haytā** *“be”*
`n`	`cop|gen:fm&num:sg`
`1|3|Aagr`	`2|3|Xcop`

ʔomēret *“say”*	*le-* *“to”*	*ʔābaʔ* *“dad”*
`part|gen:fm&num:sg`	`prep`	`n`
`3|0|Root`	`4|3|Anonagr`	`5|4|Aprep`

“Mom would have said to Dad.”

ArgumentOfExistential [**Aexs**] Identifies a relation between an existential element and a nominal or adjectival predicate. Existential elements include *yeš* “there_is”, its negative counterpart *ʔeyn* “there_is_not” and its past and future inflections of *hayā* “be”. We identify two types of existential utterances:

1. Constructions in the existential sense. The existential marker (in one of the forms described above) is the head of a nominal or adjectival predicate.

   In the following example, **ʔeyn** is the head and **yarōq** is its dependent in an `Aexs` relation:

(A.22)

ʔavāl *“but”*	*kaʔn* *“here”*	**ʔeyn** *“is_not”*	**yarōq** *“green”*
`conj`	`adv`	`exs`	`adj`
`1|3|Com`	`2|3|Madv`	`3|0|Root`	`4|3|Aexs`

“But there is no green here.”

Similarly to the Acop relation and following the discussion in 6.1, here a second analysis is also possible, where the nominal predicate is the head of the subject, and thus the head of the existential marker in a `Xexs` relation.

In the following example, **yarōq** is the head and **ʔeyn** is the dependent in a `Xexs` relation:

(A.23)	*ʔavāl “but”*	*kaʔn “here”*	**ʔeyn** *“is_not”*	**yarōq** *“green”*
	`conj`	`adv`	`exs`	`adj`
	`1|4|Com`	`2|3|Madv`	`3|4|Xexs`	`4|0|Root`

“But there is no green here.”

2. Constructions in the possessive sense, where the existential element is commonly followed or preceded by a prepositional phrase headed by the preposition *le-* “to” or *be-* “in”. We include here existential constructions with a nominal predicate and existential constructions with an adjectival predicate — the latter omit the existential element in the present tense.

In the following example, **yeš** is the head and **ʔāfro** is the dependent in an `Aexs` relation:

(A.24)	**yeš** *“there_is”*	*le- “to”*	*huʔ “he”*	**ʔāfro** *“afro”*
	`exs`	`prep`	`pro:person`	`n`
	`1|0|Root`	`2|1|Anonagr`	`3|2|Aprep`	`4|1|Aexs`

“He has an afro.”

A second analysis is also possible, where the nominal predicate is the head and the existential marker is its dependent, in a `Xexs` relation.

In the following, **ʔāfro** is the head and **yeš** is its dependent in a `Xexs` relation:

(A.25)	**yeš** *“there_is”*	*le- “to”*	*huʔ “he”*	**ʔāfro** *“afro”*
	`exs`	`prep`	`pro:person`	`n`
	`1|4|Xexs`	`2|1|Anonagr`	`3|2|Aprep`	`4|0|Root`

“He has an afro.”

A.2 Modifiers

We now move on to the second group of relations, *Modifiers*. Recall that we define modifiers as words which select their heads, express some property of it and can occur zero or more times.

Mdet Specifies a relation between a determiner and a noun. The noun is the head and the determiner is the dependent, as in the following example, where **ʔōzen** is the head and **ha-** is the dependent in a `Mdet` relation:

(A.26) **ha-** *"the"* **ʔōzen** *"ear"*
`det` `n`
`1|2|Mdet` `2|0|Root`
"The ear."

Madj Specifies a relation between an adjective and a noun.

Mpre Specifies a relation between a dependent preposition and a head noun or a verb, where the prepositional phrase — headed by the preposition — is a modifier of the noun or verb.

In the following example, **masrēq** is the head and **ʔaxēr** is its modifying adjective. Also, **tistarqi̅** is the head and **be-** is the dependent in a `Mpre` relation:

(A.27) **be-** *"in"* **masrēq** *"comb"* **ʔaxēr** *"different"*
`prep` `n|gen:ms&num:sg` `adj|gen:ms&num:sg`
`1|4|Mpre` `2|1|Aprep` `3|2|Madj`
tistarqi̅ *"you-comb_oneself"*
`v`
`4|0|Root`
"Comb your hair with a different comb."

Mposs Specifies a relation between a noun and a subsequent possessive marker, noted by the token *'šel'*, headed by the noun. The relation between the possessive marker and the possessor following it is marked with an `Aprep` relation, and this analysis mirrors the one given for other prepositions (see Example A.27). The possessive marker is generally considered a preposition, but its unique distribution and meaning prompted naming this relation differently compared to the label when other modifying prepositions are dependent on a verb or a noun (i.e., `Mpre`).

In the following example, **cad** is the head and **šel** is the dependent in a Mposs relation. Also, **šel** is the head and **ʔanī** is its dependent in an Aprep relation:

(A.28)

ʔavāl “but”	*ze “this”*	*ha- “the”*
conj:coord	pro:den|gen:ms&num:sg	det
1|4|Com	2|4|Aagr	3|4|Mdet

cad *“side”*	**šel** *“of”*	**ʔanī** *“I”*
n|gen:ms&num:sg	prep	pro:person
4|0|Root	5|4|Mposs	6|5|Aprep

“But this is my side.”

As mentioned in section 2.2.2, in cases where the possessive is a nominal suffix (e.g. *ʔaxotī* “my sister”), we would like to split the word into three tokens (e.g. *ʔaxot* “sister” *šel* “of” and *ʔani* “I”). Thus, a Mposs relation appears following the insertion of the possessive marker.

In the following example, the noun **yēled** is the head and **šel** is the dependent in a Mposs relation. Also, **šel** is the head and the pronoun **ʔani** is the dependent in a Aprep relation, similarly to the relation between a preposition and its argument:

(A.29)

ʔīmaʔ “mother”	*ʔamrā “say”*	*šalōm “hello”*
n|gen:ms&num:sg	v|gen:fm&num:sg	co
1|2|Aagr	2|0|Root	3|2|Anonagr

le- “to”	**yēled** *“child”*	**šel** *“of”*	**ʔani** *“I”*
prep	n	prep	pro:person
4|2|Anonagr	5|4|Aprep	6|5|Mposs	7|6|Aprep

“Mom said hello to my child.”

Mnoun Specifies a relation between two nouns, commonly comprising a construct phrase (referred to in Hebrew as ‘smikhut’) often denoting possession or a performer of an action, amongst other meanings. The first noun in the construct phrase often has a distinct form, as opposed to the first noun in a noun+adjective phrase or in appositions (Glinert, 1989). The first noun is the head and the second (modifier) noun is the dependent. Note that when the two nouns comprise a single idiomatic unit they are marked as one token in the corpora.

In the following example, **cel** is the head and **ʕec** is the dependent in a Mnoun relation:

(A.30)

holkīm “walk”	*lanūax “rest”*	*be- “in/at”*	**cel** *“shadow”*
`v`	`v`	`prep`	`n`
`1|0|Root`	`2|1|Ainf`	`3|2|Mpre`	`4|3|Aprep`
ha- “the”	**ʕec** *“tree”*		
`det`	`n`		
`5|6|Mdet`	`6|4|Mnoun`		

“Going to rest in the tree’s shadow.”

In the following example, **dōda** is the head and **Xāna** is the dependent in a `Mnoun` relation:

(A.31)

dōda *“aunt”*	**Xāna** *“Hanna”*	*, “,”*	*ken “yes”*
`n`	`n:prop`	`,`	`co`
`1|0|Root`	`2|1|Mnoun`	`3|1|Punct`	`4|1|Com`

“Aunt Hanna, yes.”

Another possible use of the relation `Mnoun` is when a co-reference appears in the utterance. This occurs when two nouns refer to the same entity and one of them is a resumptive apposition. Most commonly the shared instance is a person where the undetached reference is a pronoun. The resumptive apposition is the dependent.

In the following example, **huʔ** is the head and **yēled** is the dependent in a `Mnoun` relation:

(A.32)

ma “what”	**huʔ** *“he”*		
`que`	`pro:person|gen:ms&num:sg`		
`1|3|Anonagr`	`2|3|Aagr`		
ʕoṣē “do”	*ha- “the”*	**yēled** *“child”*	*ha_ze “this”*
`part|gen:ms&num:sg`	`det`	`n|gen:ms&num:sg`	`pro:dem`
`3|0|Root`	`4|5|Mdet`	`5|2|Mnoun`	`6|5|Madj`

“What is he doing, this child?”

Madv Specifies a relation between a dependent adverbial modifier and a verb.

In the following example, **ʕoṣīm** is the head and **ʕaḳšāyw** is the dependent in a `Madv` relation:

(A.33)

ma “what”	**ʕoṣīm** *“do”*	**ʕaḳšāyw** *“now”*
`que`	`part`	`adv`
`1|2|Anonagr`	`2|0|Root`	`3|2|Madv`

“What do we do now?”

Mneg Specifies a negation of a verb or a noun. Negation words include *‘loʔ’*, *‘ʔal’* and others. The noun or the verb is the head and the negation word is the dependent.

In the following example, **makīr** is the head and **loʔ** is the dependent in a `Mneg` relation:

(A.34)

ʔanī “I”	**loʔ** *“no”*	**makīr** *“recognize”*
pro:person|num:sg	neg	part|num:sg
1|3|Aagr	2|3|Mneg	3|0|Root

sipūr “story”	*ʕal “on”*	*yanšūf “owl”*
n	prep	n
4|3|Anonagr	5|4|Mpre	6|5|Aprep

“I don’t know a story about an owl.”

In the following example, **targīz** is the head and **ʔal** is the dependent in a `Mneg` relation:[1]

(A.35)

Asaf “Asaf”	, *“,”*	**ʔal** *“don’t”*	**targīz** *“annoy”*
n:prop	,	neg	v
1|4|Voc	2|4|Punct	3|4|Mneg	4|0|Root

ʔet “ACC”	*Siwān “Siwān”*
acc	Siwān
5|4|Anonagr	6|5|Aprep

“Asaf, don’t annoy Sivan.”

Mquant Specifies a relation between most commonly a noun and a nominal quantifier, headed by the noun. `Mquant` is also specified when the noun that the quantifier relates to is headed by a preposition — in that case the head of the quantifier is the preposition and not the noun.

In the following example, **bēṭen** is the head and **raq** is the dependent in a `Mquant` relation:

(A.36)

yeš “there_is”	*le- “to”*	*ʔatā “you”*	**raq** *“only/just”*
adv	prep	pro:person	qn
1|0|Root	2|1|Anonagr	3|2|Aprep	4|5|Mquant

bēṭen *“stomach”*
n
5|1|Aagr

“You only have a stomach.”

[1] Note that this example only presents the first analysis possible for ʔet-phrases as presented in Example 6.12, i.e. ʔet is the head of the ʔet-phrase.

Modifiers, like arguments, can be also clausal:

Msub Specifies a relation between a nominal element and a relativizer of a relative clause, headed by the nominal element. The relativizer is most commonly *še-* that can be replaced by *ʔašēr* (as opposed to the case where *še-* is the complementizer of a verb and cannot be replaced by *ʔašēr*). Note that the main predicate of the subordinate clause is marked as the dependent of the relativizer with a `RelCl` relation. This is analogous to the analysis of finite clausal non-agreeing arguments, as exhibited in Example A.11.

In the following example, **balōn** is the head and **še-** is the dependent in a `Msub` relation. Also, **še-** is the head and **hitpocēc** is the dependent in a `RelCl` relation:

(A.37)

balōn *"balloon"*	**še-** *"that"*	**hitpocēc** *"explode"*
`n|gen:ms&num:sg`	`conj:subor`	`v|gen:ms&num:sg`
`1|0|Root`	`2|1|Msub`	`3|2|RelCl`

"A balloon that exploded."

Clauses can also serve as modifiers of verbs, as in the case of adverbial clauses. In these cases the relation between the main predicate of the clause and the complementizer is marked with the `SubCl` relation — similarly to the case where the clause is an argument of the verb.

In the following example, **ṣāmta** is the head and **še-** is its dependent in a `Msub` relation. Also, **še-** is the head and **yihiyē** is the dependent in a `SubCl` relation:

(A.38)

ṣāmta *"put"*	*māyim* *"water"*	*xamīm* *"hot"*
`v`	`n|gen:ms&num:mass`	`adj|gen:ms&num:pl`
`1|0|Root`	`2|1|Anonagr`	`3|2|Madj`

ʕal *"on"*	*ha-* *"the"*	*guf* *"body"*	**še-** *"that"*	*loʔ* *"no"*
`prep`	`det`	`n`	`conj:subor`	`neg`
`4|1|Anonagr`	`5|6|Mdet`	`6|4|Aprep`	`7|1|Msub`	`8|9|Mneg`

yihiyē *"be/exist"*	*le-* *"to"*	*ʔatā* *"you"*
`v|gen:ms&num:sg`	`prep`	`pro:person`
`9|7|SubCl`	`10|9|Anonagr`	`11|10|Aprep`

qar *"cold"*
`adj|gen:ms&num:sg`
`12|9|Aexs`

"You put hot water on the body so you won't be cold."

A.3 Other Relations

Voc Specifies a vocative. We identify a vocative as a named entity which relates to another speaker in the conversation, most commonly followed by a question or request in the second person. We relate a vocative to the entire utterance, so the main predicate (the root) of the utterance is the head of the vocative and the vocative is the dependent. This follows the definition of this relation in the English scheme (Sagae et al., 2010).

In the following example, **tedabēr** is the head and **Asaf** is the dependent in a `Voc` relation:

(A.39)

Asaf *"Asaf"*	, *","*	**tedabēr** *"speak"*
`n:prop`	`,`	`v`
`1|3|Voc`	`2|3|Punct`	`3|0|Root`

"Asaf, speak up."

Com Specifies a communicator. Communicators include discourse markers such as *ʔavāl* "but", *ʔaz* "so", *kāḳa* "like_that", *ken* "yes" and others as well as verbs such as *tirʔē* "look" and *bōʔi* "come_here". Similarly to `Voc`, the main predicate (the root) of the utterance is the head of the relation and the communicator is the dependent. The main difference between the two relations is that `Com` does not include named entities.

In the following example, **nafāl** is the head and **ʔaz** is the dependent in a `Com` relation:

(A.40)

ʔaz *"so"*	*huʔ "he"*	**nafāl** *"fall"*
`adv`	`pro:persom|gen:ms&num:sg`	`v|gen:ms&num:sg`
`1|3|Com`	`2|3|Aagr`	`3|0|Root`

mi- "from"	*po "adv"*
`prep`	`adv`
`4|3|Mpre`	`5|4|Aprep`

"So he fell from here."

Coordination [Coord] Specifies a coordination relation between coordinated items and conjunctions, most commonly *we-* "and". Note that in the standard orthography, the coordinators are attached to the following word orthographically, but in our transcription they are separated and are considered a separate token.

There are two main approaches to dealing with coordination and coordinated elements. We follow the solution proposed in the English CHILDES scheme (Sagae et al., 2010): The English scheme adopts the approach that the head of the construction is the coordinating conjunction, the dependents are the conjuncts and the relation between the coordinator and the conjuncts is marked Coord. The coordinating conjunction represents the conjunts in the utterance, thus it is dependent with the relation the conjucts would have dependent with had they appeared separately. If there are two or more elements of coordination with multiple coordinators, the coordinators are linked from left to right (the right-most coordinator is the head of the others) by a Coord relation. In the absence of the coordinator the right-most conjunct is the head of the relation.

In the following example, **we-** is the head and **hayā** and **gdolā** are its dependents in a Coord relation:[2]

```
(A.41) tagīdi "say"  ze "this"                    hayā "be"
       v             pro:dem|gen:ms&num:sg        v|gen:ms&num:sg
       1|7|Com       2|3|Aagr                     3|7|Coord
     kše- "when"     hayīt "be"         qṭanā "small"        we- "and"
     conj:subor      v|gen:fm&num:sg    adj|gen:fm&num:sg    conj
     4|3|Msub        5|4|SubCl          6|5|Acop             7|0|Root
     ʕaḳšāyw "now"   ʔat "you"
     adv             pro:person|gen:fm&num:sg
     8|10|Madv       9|10|Aagr
     gdolā "big"
     adj|gen:fm&num:sg
     10|7|Coord
```

"Tell me, this was when you were little and now you are grown up?"

In the following example, **ʔavāl** is the head and **ṭov** is its dependent in a Coord relation in its two occurrences:

```
(A.42) yeš "there_is"  le- "to"        ʔanī "I"       zikarōn "memory"
       adv             prep            pro:person     n|gen:ms&num:sg
       1|0|Root        2|1|Anonagr     3|2|Aprep      4|1|Aexs
     ṭov "good"          ʔavāl "but"   loʔ "no"
     adj|gen:ms&num:sg   conj:coord    neg
     5|6|Coord           6|4|Madj      7|8|Mneg
     ṭov "good"          meʔōd "very"
     adj|gen:ms&num:sg   adv
     8|6|Coord           9|8|Madj
```

[2] Note that this example only presents the first possible analysis of copula constructions in which the copula element is the head of the construction, as exhibited in Example A.17.

“I have good memory, but not very good.”

In the following example, **we-** is the head and **kaʔāv** is the dependent in a Coord relation in its two occurrences:

(A.43)

le- *“to”*	*ʔat* *“you”*	**kaʔāv** *“hurt”*	*ha* *“the”*
prep	pro:person	v|gen:ms&num:sg	det
1|3|Anonagr	2|1|Aprep	3|6|Coord	4|5|Mdet

garōn *“throat”*	**we-** *“and”*	*le-* *“to”*	*ʔanī* *“I”*
n|gen:ms&num:sg	conj	prep	pro:person
5|3|Aagr	6|0|Root	7|9|Anonagr	8|7|Aprep

kaʔāv *“hurt”*	*ha-* *“the”*	*ʔoz̄en* *“ear”*
v|gen:ms&num:sg	det	n|gen:fm&num:sg
9|6|Coord	10|11|Mdet	11|9|Aagr

“You were hurt in the throat and I was hurt in the ear.”

In the following example, **we-** is the head and **rac** and **hitxabēʔ** are its dependents in a Coord relation:

(A.44)

huʔ *“he”*	**rac** *“run”*	**we-** *“and”*
pro:person|gen:ms&num:sg	v|gen:ms&num:sg	conj
1|3|Aagr	2|3|Coord	3|0|Root

hitxabēʔ *“hide”*	*meʔaxorēy* *“behind”*	*ha-* *“the”*	*ʕec* *“tree”*
v|gen:ms&num:sg	prep	det	n
4|3|Coord	5|4|Mpre	6|7|Mdet	7|5|Aprep

“He ran and hid behind the tree.”

In the following example, **we-** is the head and **ʔet** is its dependent in a Coord relation in its two appearances:[3]

(A.45)

loʔ *“no”*	*ʔat* *“you”*	*loʔ* *“no”*
neg	pro:person|gen:fm&num:sg	neg
1|4|Com	2|4|Aagr	3|4|Mneg

makirā *“recognize”*	*loʔ* *“no”*	**ʔet** *“ACC”*	*ʔarye* *“Arye”*
part|gen:fm&num:sg	neg	acc	n:prop
4|0|Root	5|6|Mneg	6|8|Coord	7|6|Aprep

we- *“and”*	*loʔ* *“no”*	**ʔet** *“ACC”*	*ʔeliyāhu* *“Eliyahu”*
conj	neg	acc	n:prop
8|4|Anonagr	9|10|Mneg	10|8|Coord	11|10|Aprep

“No, you don’t know neither Arie nor Eliyahu.”

[3] Note that this example only presents the first possible analysis of ʔet-phrases where ʔet is the head and the subsequent noun is its dependent, as exhibited in Example 6.12.

Serialization [Srl] Specifies a serial verb. A serial verb relation occurs between two finite verbs occurring sequentially in the same clause with no conjunction between them. The first verb is the dependent and the second verb is the head. The first verb is in the imperative mood. We follow here the definition of this relation in Sagae et al. (2010), though it is worth noting that this is not the exact theoretic definition of serial verbs. For example, serial verbs refer to two verbs of the same subject, whereas in this case serial verbs can refer to two different subjects.

In the following example, **nevaqēr** is the head and **bōʔi** is the dependent:[4]

(A.46)	**bōʔi** *"come"*	**nevaqēr** *"visit"*		
	`v|form:imp&gen:fm&num:sg`	`v`		
	`1|2|Srl`	`2|0|Root`		
	maxār "tomorrow"	*ʔet "ACC"*	*ʔīmaʔ "mother"*	*šel "of"*
	`adv`	`acc`	`n`	`prep`
	`3|2|Madv`	`4|2|Anonagr`	`5|4|Aprep`	`6|5|Mposs`
	hiʔ "she"			
	`pro:person`			
	`7|6|Aprep`			

"Let's visit her mother tomorrow."

Enumeration [Enum] Specifies an enumeration relation. An enumeration is a sequence of numbers, words or letters with no explicit coordination, such as *ʔaxāt* "one", *štāyim* "two", *šalōš* "three". The head is the last token in the sequence and the rest of the tokens are dependent on it with an `Enum` relation.

In the following example, **šeš** is the head and **ʔaxāt**, **štāyim**, **šalōš**, **ʔārbaʕ** and **xamēš** are all its dependents in an `Enum` relation:

(A.47)	**ʔaxāt** *"one"*	, *","*	**štāyim** *"two"*	, *","*
	`num`	`,`	`num`	`,`
	`1|11|Enum`	`2|11|Punct`	`3|11|Enum`	`4|11|Punct`
	šalōš *"three"*	, *","*	**ʔārbaʕ** *"four"*	, *","*
	`num`	`,`	`num`	`,`
	`5|11|Enum`	`6|11|Punct`	`7|11|Enum`	`8|11|Punct`

[4] Note that this example only presents the first possible analysis of ʔet-phrases where ʔet is the head and the subsequent noun is its dependent, as exhibited in Example 6.12.

xamēš *"five"*	, ","	**šeš** *"six"*
`num`	`,`	`num`
`9|11|Enum`	`10|11|Punct`	`11|0|Root`

"One, two, three, four, five, six."

Unknown [Unk] Specifies an unclear or unknown word — most commonly a child invented word — which appears disconnected from the rest of the utterance and acts often as filler syllables. A child invented word is a word undefined in the dictionary and seems to be a form invented by the child, or a word indicating the child babbles or speaks incoherently. It is represented in the mor tier with the 'chi' part of speech. When an unknown word is uttered with no apparent connection to the rest of the utterance, we mark it as a dependent of the main predicate of the utterance with a `Unk` relation.

In the following example, **naʕaṣe** is the head and **e** is its dependent in an `Unk` relation:

(A.48)

boʔ "come"	**naʕaṣe** *"do"*	*parcūf "face"*	*šel "of"*	**e** *"e"*
`v`	`v`	`n`	`prep`	`chi`
`1|2|Srl`	`2|0|Root`	`3|2|Anonagr`	`4|3|Mposs`	`5|2|Unk`
ṣaqīt "bag"				
`n`				
`6|4|Aprep`				

"Let's make a face of a bag."

When the unknown word or child invented word takes the place of a known relation, it is marked with that relation and not with the `Unk` relation.

In the following example, **le-** is the head and **bdibiyabi** is the dependent in an `Aprep` relation, being the argument of a preposition which clearly takes the place of a location:

(A.49)

ʕaḳšāyw "now"	*ʔanī "I"*	*holēḳet "walk"*	**le-** *"to"*
`adv`	`n|num:sg`	`part|num:sg`	`prep`
`1|3|Madv`	`2|3|Aagr`	`3|0|Root`	`4|3|Anonagr`
bdibiyabi *"bdibiyabi"*			
`chi`			
`5|4|Aprep`			

"Now I am going to bdibiyabi."

Punctuation [Punct] Specifies a punctuation mark. Punctuations can be either utterance-final punctuation marks or commas. Punctuation marks were added to the transcriptions to highlight the tone of the utterances. Punctuation marks are dependent on the root of the utterance with the `Punct` relation.

In the following example, the comma and the dot are marked with the `Punct` relation. In this example we also include the utterance-final punctuation mark (the dot) whereas in all other examples we did not include it for ease of reading:

(A.50)

ʔat "you"	*loʔ "no"*	**criḳā** *"necessary"*	
`pro:person|gen:fm&num:sg`	`neg`	`adj|gen:fm&num:sg`	
`1|3|Aagr`	`2|3|Mneg`	`3|0|Root`	
lefaxēd "be_scared"	**,** *","*	*xamudā "cute"*	**.** *"."*
`v`	`,`	`n|gen:fm&num:sg`	`.`
`4|3|Ainf`	`5|3|Punct`	`6|3|Voc`	`7|3|Punct`

"You shouldn't be scared, sweety."

Elision relations Utterances may exhibit ellipsis of certain tokens. Following the English scheme (Sagae et al., 2010), we use elision relations to mark the relations that would have been annotated if the elided element was present. This type of relation is a concatenation of two relations — the first relation is the relation that would have been marked between the existing token and the elided token; and the second relation is the relation that would have been marked between the elided token and its head.

In the following example, **ʔet** is the head and **ha-** is its dependent in a `Mdet-Aprep` relation — `Mdet` to mark **ha-** the determiner of an elided element, and `Aprep` to mark that the elided element would have taken the place of the argument of **ʔet** (annotated here with the approach that treats it similarly to a preposition):

(A.51)

hīne "here"	*bubā "doll"*	**ʔoḳēlet** *"eat"*
`co`	`n|gen:fm&num:sg`	`part|gen:fm&num:sg`
`1|3|Com`	`2|3|Aagr`	`3|0|Root`
ʔet "ACC"	**ha-** *"the"*	
`acc`	`det`	
`4|3|Anonagr`	`5|4|Mdet-Aprep`	

"Here, a doll is eating the —"

In the following example, **hiʔ** is marked with the `Aagr-Root` relation — `Aagr` to mark **hiʔ** as the agreeing argument of the elided element, and `Root` to mark that the elided element would have taken the place of the main predicate (root) of the utterance:

(A.52)	*ʔoy “oh_no”*	, “,”	*kāma “how_much”*	**hiʔ** *“she”*
	`co`	`,`	`que`	`pro:person`
	`1|4|Com`	`2|4|Punct`	`3|4|Mquant`	`4|0|Aagr-Root`

“Oh no, she is so —”

Bibliography

Alfred V. Aho and Jeffrey D. Ullman. *The theory of parsing, translation, and compiling.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1972. ISBN 0-13-914556-7.

Aviad Albert, Brian MacWhinney, Bracha Nir, and Shuly Wintner. The Hebrew CHILDES corpus: Transcription and morphological analysis. *Language Resources and Evaluation*, forthcoming. Accepted for publication.

Miguel Ballesteros and Joakim Nivre. MaltOptimizer: A system for MaltParser optimization. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7.

Ruth A. Berman. *Modern Hebrew Structure.* University Publishing Projects, Tel Aviv, 1978.

Ruth A. Berman and Jürgen Weissenborn. *Acquisition of word order: A crosslinguistic study.* German-Israel Foundation for Research and Development (GIF), Jerusalem, Israel, 1991. HEBREW.

James P. Blevins. Word-based morphology. *Journal of Linguistics*, 42(4): 531–573, 2006. doi: 10.1017/S0022226706004191.

Gabi Danon. Caseless nominals and the projection of DP. *Natural Language & Linguistic Theory*, 24(4):977–1008, 2006. doi: 10.1007/s11049-006-9005-6. URL `http://dx.doi.org/10.1007/s11049-006-9005-6`.

Marie-Catherine de Marneffe and Christopher D. Manning. Stanford typed dependencies manual. 2008a. URL `http://nlp.stanford.edu/software/dependencies_manual.pdf`.

Marie-Catherine de Marneffe and Christopher D. Manning. The Stanford typed dependencies representation. In *COLING Workshop on Cross-framework and Cross-domain Parser Evaluation*, 2008b. URL `pubs/dependencies-coling08.pdf`.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. Generating typed dependency parses from phrase structure trees. In *LREC*, 2006. URL `http://nlp.stanford.edu/pubs/LREC06_dependencies.pdf`.

Gülşen Eryiğit, Joakim Nivre, and Kemal Oflazer. Dependency parsing of Turkish. *Computational Linguistics*, 34(3):357–389, 2008. ISSN 0891-2017. doi: http://dx.doi.org/10.1162/coli.2008.07-017-R1-06-83.

Shadi Ganjavi. *Direct Objects in Persian*. University of Southern California, 2007. ISBN 9780549004097. URL `http://books.google.co.il/books?id=qK85twAACAAJ`.

Lewis Glinert. *The Grammar of Modern Hebrew*. Cambridge University Press, 1989. ISBN 9780521611886. URL `http://books.google.co.il/books?id=wKuEtVhnCpkC`.

Yoav Goldberg. *Automatic Syntactic Processing of Modern Hebrew*. PhD thesis, Ben Gurion University of the Negev, Israel, 2011.

Spence Green and Christopher D. Manning. Better arabic parsing: baselines, evaluations, and analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 394–402, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1873781.1873826`.

Nizar Habash and Ryan M. Roth. CATiB: The Columbia Arabic treebank. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 221–224, 2009.

Jan Hajič and Petr Zemánek. Prague Arabic dependency treebank: Development in data and tools. In *Proceedings of the NEMLAR Intern. Conf. on Arabic Language Resources and Tools*, pages 110–117, 2004.

Jan Hajič, Eva Hajičová, Petr Pajas, Jarmila Panevová, Petr Sgall, and Barbora Vidová Hladká. Prague Dependency Treebank 1.0 (Final Production Label), 2001.

Eva Hajičová, Zdeněk Kirschner, and Petr Sgall. A Manual for Analytic Layer Annotation of the Prague Dependency Treebank (English translation). Technical report, ÚFAL MFF UK, Prague, Czech Republic, 1999.

Sandra Kübler, Ryan T. McDonald, and Joakim Nivre. *Dependency Parsing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2009.

Richard Kunert, Raquel Fernández, and Willem Zuidema. Adaptation in child directed speech: Evidence from corpora. In *SemDial 2011: Proceedings of the 15th Workshop on the Semantics and Pragmatics of Dialogue*, pages 112–119, 2011.

Brain MacWhinney. *The CHILDES Project: Tools for Analyzing Talk*. Lawrence Erlbaum Associates, Mahwah, NJ, 2000.

Yuval Marton, Nizar Habash, and Owen Rambow. Dependency Parsing of Modern Standard Arabic with Lexical and Inflectional Features. *Computational Linguistics*, 39(1), 2013.

Yael Dahan Netzer and Michael Elhadad. Generation of noun compounds in hebrew: Can syntactic knowledge be fully encapsulated? In *Proceedings of the Ninth International Workshop on Natural Language Generation*, pages 168–177, 1998.

Anat Ninio. A proposal for the adoption of dependency grammar as the framework for the study of language acquisition. In G. Ben Shakhar and A. Lieblich, editors, *Volume in Honor of Shlomo Kugelmass*, pages 85–103. Jerusalem: Magnes, 1996.

Anat Ninio. Acquiring a dependency grammar: The first three stages in the acquisition of multiword combinations in hebrew-speaking children. In G. Makiello-Jarza, J. Kaiser, and M. Smolczynska, editors, *Language acquisition and developmental psychology*. Carcow: Universitas, 1998.

Bracha Nir and Ruth A. Berman. Parts of speech as constructions: The case of hebrew adverbs. *Constructions and Frames*, 2(2):242–274, 2010. doi: doi:10.1075/cf.2.2.05nir. URL `http://www.ingentaconnect.com/content/jbp/cf/2010/00000002/00000002/art00006`.

Bracha Nir, Brian MacWhinney, and Shuly Wintner. A morphologically-analyzed CHILDES corpus of Hebrew. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA). ISBN 2-9517408-6-7.

Joakim Nivre, Johan Hall, and Jens Nilsson. MaltParser: A Data-Driven Parser-Generator for Dependency Parsing. In *Proceedings of the fifth international conference on Language Resources and Evaluation (LREC2006), May 24-26, 2006, Genoa, Italy*, pages 2216–2219. European Language Resource Association, Paris, 2006.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. The CoNLL 2007 Shared Task on Dependency Parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL*, pages 915–932, Prague, 2007.

W. Keith Percival. On the historical source of immediate constituent analysis. In James D. McCawley, editor, *Syntax and Semantics Volume 7, Notes from the Linguistic Underground*, pages 229–242. Academic Press, New York, 1976.

Haim B. Rosen. *Ivrit Tova ("Good Hebrew")*. Kiryat Sefer, Jerusalem, Israel, second edition, 1966. HEBREW.

Kenji Sagae and Alon Lavie. A best-first probabilistic shift-reduce parser. In *Proceedings of the COLING/ACL on Main conference poster sessions*, COLING-ACL '06, pages 691–698, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1273073.1273162.

Kenji Sagae and Jun'ichi Tsujii. Dependency parsing and domain adaptation with LR models and parser ensembles. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 1044–1050, June 2007. URL http://www.aclweb.org/anthology/D/D07/D07-1111.

Kenji Sagae, Eric Davis, Alon Lavie, Brian MacWhinney, and Shuly Wintner. Morphosyntactic annotation of CHILDES transcripts. *Journal of Child Language*, 37(3):705–729, 2010. doi: 10.1017/S0305000909990407. URL http://journals.cambridge.org/article_S0305000909990407.

Khalil Sima'an, Alon Itai, Yoad Winter, Alon Altman, and N. Nativ. Building a tree-bank of Modern Hebrew text. *Traitment Automatique des Langues*, 42(2), 2001.

Otakar Smrž and Petr Pajas. *MorphoTrees of Arabic and Their Annotation in the TrEd Environment*, pages 38–41. ELDA, 2004.

Reut Tsarfaty and Yoav Goldberg. Word-based or morpheme-based? annotation strategies for Modern Hebrew clitics. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. European Language Resources Association (ELRA), May 2008. ISBN 2-9517408-4-0. URL `http://www.lrec-conf.org/proceedings/lrec2008/`.

Reut Tsarfaty, Joakim Nivre, and Evelina Andersson. Evaluating dependency parsing: Robust and heuristics-free cross-annotation evaluation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 385–396, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D11-1036`.

ניתוח תחבירי של קורפוס CHILDES בעברית

שי גרץ

ניתוח תחבירי של קורפוס CHILDES בעברית

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר מגיסטר למדעים

במדעי המחשב

שי גרץ

הוגש לסנט הטכניון – מכון טכנולוגי לישראל

סיון תשע"ג חיפה מאי 2013

המחקר נעשה בהנחיית פרופ’ אלון איתי (הפקולטה למדעי המחשב, הטכניון) ופרופ’ שולי וינטנר (הפקולטה למדעי המחשב, אוניברסיטת חיפה) בפקולטה למדעי המחשב.

אני מודה לטכניון על התמיכה הכספית הנדיבה בהשתלמותי.

תקציר

מאגר CHILDES הינו מאגר של תמלילים של שיחות בין ילדים ומבוגרים ביותר מ־25 שפות. המאגר משמש כלי חשוב לחוקרים בנושאים של רכישת שפה והתפתחות שפה בקרב ילדים. הוא עושה זאת על ידי הצגה של התמלילים בשכבות מידע שכוללות פונולוגיה, מורפולוגיה ותחביר. בעבר כאשר חוקרים ביקשו לבחון התפתחות מבנים תחביריים מסוימים, הם נאלצו לתייג באופן ידני חלקים מהמאגר לצרכיהם. שיטה זו הייתה בזבזנית ויקרה שכן כל חוקר נאלץ להשקיע זמן רב בהכנת המידע הדרוש לו. כמו כן, היה קשה יותר לשתף בין החוקרים את המידע המתויג. בעבודה זו אנו מתארים תיוג תחבירי ממוחשב של תמלילי CHILDES בעברית.

אחד הכלים האוטומטיים שיכולים לפתור בעיה זו הוא מנתח תחבירי (Parser) – אלגוריתם שמשרה ניתוח תחבירי על משפט באופן אוטומטי. בשפות רבות פותחו מנתחים תחביריים והם מותאמים לשפה ולז'רגון הספציפי שעל פיו הם נבנו. בפרט, בעשור האחרון נעשה שימוש הולך וגובר בניתוח תחבירי מבוסס דקדוק תלויות (Dependency Parsing) – ניתוח תחבירי המשרה מבנה תלויות (מבנה המבוסס על דקדוק תלויות או Dependency Grammar) על משפט. במבנה תלויות משפט מיוצג על ידי גרף, שצמתיו הם המילים במשפט וכל קשת בין זוג מילים מייצגת יחס. בכל יחס כזה מילה אחת משמשת כ־head (השולטת על היחס) והשנייה כ־dependent ולכל מילה יש בדיוק head אחד ואפס או יותר dependents. באופן זה מבנה תלויות מאפשר להציג את היחסים בין המילים במשפט כעץ. מבנה תלויות יכול להיות מסומן (Labeled) – כאשר כל יחס בין שתי מילים מסומן עם תפקיד פונקציונלי כלשהו; או לא־מסומן (Unlabeled) כאשר רק ההיררכיה של היחסים חשובה ולא התפקיד שהם ממלאים. המילה המרכזית במשפט שאינה תלויה באף מילה אחרת הינה השורש (root) של המשפט. בעברית, מילה זו היא בדרך כלל הפועל המרכזי במשפטים פועליים או הפרדיקט השמני במשפטיים שימניים.

ניתוח תחבירי מסוג זה נמצא מועיל בייחוד בשפות בעלות סדר מילים חופשי יחסית, הוא מאפשר להציג קשרים בין מילים בצורה ברורה וקריאה ולהשתמש בכלים הלקוחים מהעולם של מערכות לומדות. בפרט, שיטה נפוצה למימוש ניתוח תחבירי מבוסס דקדוק תלויות היא Data driven parsing – מנתח תחבירי המשתמש במסווג שהתאמן על קבוצת משפטים מתויגת כהלכה כדי לסייע בפעולת הניתוח התחבירי.

תהליך העבודה של Data driven parser בנוי משני חלקים מרכזיים: 1) למידה – שלב שבו נבנה מודל על סמך קבוצה של משפטים מתויגים באופן ידני.
2) ניתוח – שלב שבו מתבצע מעבר על המילים במשפט ובכל איטרציה הפעולה הנבחרת (לעבור למילה הבאה או לסמן יחס כלשהו בין שתי מילים) נקבעת על סמך המודל שנבנה בשלב הלמידה. בשני חלקים אלו ניתן להיעזר בכל

המידע העומד לרשותנו מהמשפט הנתון. מידע זה מרכיב את קבוצת התכוניות (feature set) שעליהן מתבסס אלגוריתם הלמידה ואלגוריתם הניתוח. בפרט, תכוניות מורפולוגיות יכולות לסייע בקבלת ההחלטה. כדוגמה אפשר להסתכל על תכונת ההסכמה בעברית – שם עצם בעמדת נושא חייב להתאים לפועל במין, במספר ובגוף. התכוניות המורפולוגיות הללו יכולות להיות מכריעות בקבלת ההחלטה לנתח תחבירית את הצירוף "אפרוחים חיפש" ביחס של מושא ישיר (של "אפרוחים" כלפי "חיפש") ולא נושא.

לאחרונה פותח מנתח תחבירי כזה לאנגלית במיוחד עבור CHILDES. בעזרת המנתח התחבירי נותח אוסף קורפוסים זה במלואו תוך שימוש בקבוצת יחסי תלויות שהוגדרו ספציפית למאגר זה. בעבודה זו אנו מתארים את תהליך בנייתו של מנתח תחבירי מבוסס תלויות לעברית אשר משמש לניתוח תחבירי מלא של מאגר CHILDES בעברית. תהליך העבודה מורכב משלושה חלקים עיקריים:

1. הגדרת קבוצת היחסים (המכונה "סכמה תחבירית" או Annotation Scheme) שבעזרתה יתויג החלק העברי של CHILDES תוך מתן תשומת לב למאפייני הקורפוסים.
2. אימון מנתח תחבירי על חלק מהקורפוס שתויג ידנית וניתוח תחבירי של שאר חלקי הקורפוס.
3. הערכה אמפירית של המנתח וניסיון להתמודד עם אתגרים שהועלו במהלך בניית הסכמה התחבירית באמצעים אמפיריים, כולל השוואה בין גישות שונות להגדרת יחסים וכן בחינת התרומה של התכוניות המורפולוגיות להצלחת הניתוח.

בניית הסכמה התחבירית בעבודה זו הסתמכה על עקרון ההבחנה בין משלימים מוצרכים (Arguments) ללוואיים (Modifiers). יחסים מהקבוצה הראשונה כוללים בתפקיד ה־dependent מילים שמהוות משלים מוצרך של מילה אחרת. לדוגמה, הפועל "קנה" – מצריך משלים שמבצע את הפעולה (מי קנה) ומשלים שעליו התבצעה הפעולה (את מה קנה). שני החלקים האלו במשפט "הוא קנה אוטו" מסומנים כמשלימים מוצרכים ביחס לפועל המרכזי "קנה".

יחסים מהקבוצה השנייה כוללים בתפקיד ה־dependent מילים שהן אופציונליות ומגדירות תכונות מסוימות של ה־head שעליהן הן תלויות. לדוגמה, במשפט "הוא קנה אוטו אדום" – המילה "אדום" הינה לוואי (במקרה זה – לוואי תואר) של המילה "אוטו" שכן היא איננה מוצרכת (המשפט תקין לחלוטין בלעדיה) והיא מציינת תכונה של האוטו.

בעבודה זו אנו מציגים שיטות שונות להערכה אמפירית של איכות המנתחים התחביריים והסכמה התחבירית. במסגרת ההערכה האמפירית השתמשנו בשני מנתחים תחביריים – במנתח תחבירי בשם MEGRASP ששימש לניתוח התחבירי של מאגר CHILDES באנגלית; וב־MaltParser, מנתח תחבירי שמאפשר בין השאר להגדיר מחדש את קבוצת התכונות שמשמשות אותו בשלב הלמידה. בתהליך ההערכה האמפירית בעבודה זו אנו מראים את ההתאמה של שני המנתחים לקורפוסים של CHILDES בעברית, תוך שימוש בסכמה התחבירית

שהוגדרה. עולה כי MaltParser טוב מ־MEGRASP באופן מובהק וזאת כנראה בשל כיוונון מדויק של הפרמטרים שלו וכן קבוצת תכונות עשירה יותר. אנו מראים את יכולת ההכללה של המנתחים לניתוח תחבירי של משפטים הלקוחים מקורפוס שונה מזה שהתאמנו עליו. אנו מבחינים בין קונפיגורציות שונות של למידה ובחינה – למידה ובחינה של המנתח על משפטים של ילדים בלבד (CS - Child Speech); למידה ובחינה של המנתח על משפטים שנאמרו על ידי מבוגרים לילדים (CDS - Child-directed Speech); ולמידה של המנתח על משפטים שנאמרו על ידי מבוגרים לילדים ובחינה שלו על משפטים של ילדים. אנו דנים בהבדלים המשמעותיים בדיוק של המנתח בין הקונפיגרציות הללו. הבדלים מן הסוג הזה יכולים לתרום להבנה עמוקה יותר של תהליך רכישת השפה אצל ילדים.

כמו כן, בעבודה זו אנו מראים את השיפור בדיוק של הניתוח התחבירי לאחר שימוש בתכוניות מורפולוגיות הנתונות בשכבת המידע המורפולוגית של CHILDES. אנו מראים כיצד הוספה של התכוניות המורפולגיות "מין", "מספר" ו"גוף" משפרת את הדיוק של המנתח התחבירי, אם כי לא באופן מובהק. כאמור, תכוניות אלו יכולות לסייע לקביעה של יחסים הקשורים להתאמה כמו יחס בין שם עצם לשם תואר או בין שם עצם בעמדת הנושא לפועל המרכזי במשפט.

תהליך בניית הסכמה התחבירית כלל הכרעות לגבי הגדרות של יחסים שאין עליהן תמימות דעים בקהילה הבלשנית. בעבודה זו אנו בוחנים חלופות למספר תופעות לשוניות שלגבי התיוג התחבירי הנכון שלהן קיימות תיאוריות שונות. אנו מתייגים את הקורפוסים שבהם השתמשנו לצורכי ההערכה בשתי הצורות ובוחנים את הדיוק של המנתח התחבירי על סמך כל אחת מהסכמות התחביריות.

תופעה לשונית אחת כזו שנבחנת בעבודה היא תופעת האוגד, שאיננה ייחודית לשפה העברית. במשפטים בהווה האוגד הוא אופציונלי ושני המשפטים "האוטו הוא יפה" ו"האוטו יפה" תקינים מבחינה תחבירית. נשאלת השאלה האם יש לסמן את האוגד "הוא" כשורש שכן הוא מהווה נטייה בזמן הווה של הפועל "היה" (וכאמור הפועל המרכזי הוא בדרך כלל השורש של המשפט), או לסמן את "יפה" כשורש שכן האוגד הוא אופציונלי וכאשר הוא איננו נמצא השורש הינו המילה "יפה" (כמו במשפט "האוטו יפה").

תופעה לשונית נוספת שנבחנת בעבודה היא סמן יחסת האקוזטיב (יחסה המסמנת מושא ישיר) "את". המילה "את" יכולה להופיע לפני שם עצם מיודע בעמדת מושא ישיר כמו במשפט "הוא קנה את האוטו". גם כאן ישנן שתי גישות בנוגע לזהות ה־head בצירוף כמו "את האוטו" – האם "את" משמש כ־head או שמא "האוטו".

בחינת התופעות הלשוניות מעלה כי לצורך הניתוח התחבירי של קורפוס CHILDES בעברית אין עדיפות מכרעת לחלופה אחת על פני האחרת. ייתכן כי מאפייניו של הקורפוס או גודלו הקטן יחסית הם שגרמו לכך שאין יתרון מובהק לאחת החלופות הלשוניות. עם זאת, ייתכן כי אין זה משנה באיזו חלופה נבחר

וכי כל עוד התיוג הוא עקבי המנתח התחבירי מצליח לתפוס במידה טובה מספיק את התנאים לתיוג מדויק של הצירופים הללו.

המנתח התחבירי שפיתחנו בעבודה זו, וכן הקורפוס המתויג (הן ידנית והן אוטומטית) עומדים לרשות החוקרים כחלק מהמאגר הפתוח של CHILDES. תקוותינו היא שהחומרים הללו ישמשו בעתיד הקרוב לצורך ניתוחים לשוניים ופסיכו־לשוניים של תהליכי שפה בקרב ילדים.